

Centro Técnico Aeroespacial – CTA
Instituto Tecnológico de Aeronáutica – ITA
Divisão de Pós-Graduação

Uma Arquitetura para Agentes Autônomos
Móveis sob a Visão da Inteligência
Artificial Distribuída

Claudio Cesar de Sá

São José dos Campos (SP), 29 de abril de 1997

Resumo

Esta tese apresenta uma arquitetura distribuída para agentes autônomos do tipo móvel. As características dessa arquitetura são estudadas pela sua instanciação na forma de três agentes computacionais, que possuem funcionalidades voltadas à exploração de “nichos”. A metodologia utilizada para testar e validar a operacionalidade desses agentes é um ambiente simulado. A arquitetura é modelada em três níveis hierárquicos. O *nível reativo* é constituído por uma rede de agentes nebulosos. Essa rede é uma estrutura distribuída, que conecta agentes sensoriais a agentes motores e é capaz de controlar a locomoção de um agente em um ambiente desconhecido, utilizando um enfoque de *Solução Distribuída de Problemas*. O *nível instintivo* é formado por um conjunto de agentes interdependentes, que dispõem de um certo conhecimento. Esse conhecimento é expresso por regras de produção e corresponde a esquemas de comportamento estereotipados, que podem ser aplicados ao nível reativo de acordo com a situação do agente no ambiente. Esses esquemas de comportamento têm inspiração biológica. O *nível cognitivo* consiste simplesmente em um *autômato comportamental*, utilizado para gerenciar os objetivos globais do agente. As características da arquitetura são: baixo custo computacional, adaptação, distribuição, inspiração biológica, robustez, flexibilidade e utilização de esquemas de comportamento baseados em contexto. O fato da arquitetura incluir duas comunidades distintas de agentes computacionais – os níveis reativo e instintivo – permite que conceitos como competição, cooperação e conflito possam ser estudados dentro de um enfoque conceitual de *Inteligência Artificial Distribuída*. Finalmente, com essa arquitetura pode-se investigar questões ligadas à análise e à síntese de comportamentos em agentes autônomos.

Abstract

This thesis presents a distributed architecture for mobile autonomous agents. The characteristics of this architecture are studied on the basis of its instantiation which is implemented in the form of three computational agents. The functionality of these agents explores the “niche space”. The methodology used is to test and validate the architecture in a simulation environment. The architecture possesses three hierarchical levels. The *reactive level* consists of a fuzzy agent network. This network is a distributed structure that connects sensorial agents to motor agents and it is able to control the agent locomotion in an unknown environment within a *Distributed Problem Solving* approach. The *instinctive level* consists of a set of interdependent agents that possess some knowledge. This knowledge takes the form of production rules and corresponds to stereotyped behavior schemes. Such behavior schemes have a biological inspiration. The *cognitive level* consists simply of a *behavior automaton* that manages the agent global goals. The characteristics of the architecture are the following: low computational cost, adaptability, distributivity, biological inspiration, robustness, flexibility and compatibility of context-based behavior schemes. The fact that the architecture includes two distinct agent communities – the reactive and instinctive level agents – leads to the study of concepts like competition, cooperation and conflict in a *Distributed Artificial Intelligence* conceptual framework. Finally, based upon this architecture, it is possible to investigate the synthesis and analysis of autonomous agent behaviors.

Tese apresentada à Divisão de Pós-graduação do Instituto tecnológico de Aeronáutica como parte dos requisitos para a Obtenção do título de Doutor em Ciências, na Área de Informática do Curso de Engenharia Eletrônica e Computação

Claudio Cesar de Sá

Uma Arquitetura para Agentes Autônomos Móveis sob a
Visão da Inteligência Artificial Distribuída

Tese aprovada na sua versão final pelos abaixo assinados

Prof. Dr. Nizam Omar
Orientador

Prof. Dr. Paulo Afonso de O. Soviero
Chefe da Divisão de Pós-Graduação

Uma Arquitetura para Agentes Autônomos Móveis sob a Visão da Inteligência Artificial Distribuída

Claudio Cesar de Sá

Composição da Banca Examinadora

Celso Massaki Hirata	PhD	IECC-ITA - Presidente
Nizam Omar	DC	IECT-ITA - Orientador
Celso de Renna e Souza	PhD	IECT-ITA
Pedro Paulo Balbi de Oliveira	PhD	UNIVAP
Guilherme Bittencourt	PhD	DI-UFSC

Dedicatória

*Ao meu pai, Antonio Cesar, in memoriam, que partiu durante esta jornada.
Um elo rompido . . .
Aos meus filhos, Álvaro e Cinthia.*

Agradecimentos

Esta é a primeira página que um dia pensei e a última que escrevo. Muitas pessoas me ajudaram na construção desta história. Para relacioná-las todas, certamente muitas páginas a mais seriam necessárias. Vou citar algumas sob o risco de ser ingrato com muitas. Mas, sem dúvida, todas estão presentes em minha lembrança.

- Inicialmente, agradeço à Universidade Federal de Alagoas e à CAPES pela liberação e pelo suporte financeiro durante o maior período deste trabalho;
 - Ao ITA por ter-me aceito como aluno e propiciado um ambiente de trabalho;
 - Ao Omar por ter-me dado autonomia para que encontrasse o meu caminho. Apesar de minhas “*caminhadas em círculos*”, senti que continuava acreditando em mim, fazendo com que atalhos fossem tomados sempre que possível. Sua postura filosófica foi estimulante a cada momento difícil que atravessei;
 - Ao Guilherme por ter sido tão receptivo em momentos tão difíceis desta jornada. Sua imensa sabedoria fez com que caminhos inexplorados se tornassem interessantes. Que sua paciência e compreensão, típicas de um Mestre Zen, ajudem outros alunos;
 - Aos meus professores do ITA e INPE, cujas lições foram consideradas a cada momento: Sandra, Clóvis, Cunha, Simoni, Celso de Renna e Márcio;
 - Aos professores do IEC/ITA, cujas conversas de corredor e trocas de idéias foram saudáveis e construtivas: Milton Sakude, Yano, Pellegrino (obrigadão pelo início de tudo), Felipe, Ney, Fábio, Hirata, Abdul e Jony;
 - Em especial ao Dr. Celso de Renna e Souza por seus comentários de estímulo e entusiasmo na proposta inicial deste trabalho. Espero ter correspondido a pelo menos alguma parte de suas idéias;
 - Aos colegas da Pós-Graduação, cujo convívio foi um grande aprendizado: Milton Abrunhosa, Dedê, Jamil, Aury, Crispin, Malaquias, Juliatto, Moura, Adriane, Ivan A todos muito obrigado;
 - Ao pessoal que tornou as coisas mais fáceis dentro do IEC: Rosa, Léa, Conceição, Mônica, Júlia e Mário;
 - Aos companheiros do meu atual departamento, Universidade do Estado de Santa Catarina/Departamento de Ciências da Computação, que suavizaram minhas tarefas nesse fim de tese: Flávio, Márcio, Wesley, Júlio e Carlos;
 - À UNICAMP, mais especificamente ao Departamento de Engenharia de Petróleo, por permitir que eu desfrutasse de seus laboratórios;
-

- Ao Dr. Armando Freitas da Rocha (IB/UNICAMP) por ter-me apresentado a biologia aplicada à computação;
- Ao professor de biologia Paulo Roland Ünger, cuja prontidão em me ajudar foi de inestimável valor;
- Aos pesquisadores que gentilmente enviaram seus trabalhos para somar nesta história: Demazeau, Brooks, Steels, Huberman, Sloman ...;
- A gentileza e os momentos de conhecimento que usufrui junto de algumas pessoas: Orlando, Elery e Pepê – cujos comentários somaram neste trabalho;
- Aos meus amigo(a)s, em sua maioria distantes, que sempre que possível me empurravam para frente;
- Aos ex-colegas de repúblicas por me suportarem em minhas aflições. Em especial ao Roberto Galski (*grande RG*);
- Ao apoio logístico e oportuno de alguns amigos: Luiz Sérgio Heizelman (CTA / ITA) e José Arnon da Silva Costa (DCC/UFAL);
- Ao pessoal que me ajudou a esfriar a cabeça neste processo de doutoramento: Danilo Pasa, a turma da natação do meio-dia no Joinville Tênis Club e aos professores de violão;
- Às eficazes cápsulas de vitaminas, gentilmente cedidas pela Próformula (Joinville – SC), que não permitiram que eu adoecesse em momento algum;
- Aos meus familiares em geral que suportaram meus “altos e baixos” durante esses anos. Aos tios Draúcio e Jeanette, que revisaram parte desta tese;
- À Esther, minha professora de português no ginásio, pelas correções finais desta tese;
- Aos tios Mazinho e Nazareth, que gentilmente me propiciaram a estadia necessária em Florianópolis;
- À Maria Luiza, minha esposa, que sem ela nada disso seria possível. Espero, agora, poder retribuir mais tempo aos seus projetos de vida.

Com certeza devo ter esquecido alguém. Desculpem-me. Contudo, fico eternamente grato a todos.

Sumário

1	Introdução	1
1.1	Inteligência Artificial	1
1.2	Agentes Autônomos	3
1.3	Arquiteturas de AA's	6
1.4	A Motivação do Tema	10
1.4.1	Arquiteturas em IA	10
1.4.2	Locomoção	12
1.4.3	Biologia	12
1.4.4	Modelo Completo	14
1.5	A Proposta da Tese	14
1.6	Contribuições da Tese	15
1.7	O Conteúdo da Tese	16
1.7.1	Os Capítulos Seguintes	16
1.7.2	Guia para o Leitor	16
2	Elementos de Inteligência Artificial	18
2.1	Introdução	18
2.2	Representação de Conhecimento	18
2.3	Sistemas de Produção	20
2.4	Raciocínio	23
2.5	Planejamento Clássico e Reativo	25
2.5.1	Planejamento Clássico	25
2.5.2	Um Exemplo de Plano	27
2.5.3	Planejamento Reativo	31
3	Apresentação à Inteligência Artificial Distribuída	33
3.1	Introdução	33
3.2	Discussão Geral	33
3.2.1	Motivações Preliminares	33
3.2.2	Quanto à Distribuição	35
3.3	Solução Distribuída de Problemas (SDP)	36
3.3.1	Um Exemplo	36
3.3.2	Resumindo	38
3.4	Sistemas Multiagentes (SMA)	38

3.5	Relação entre SMA e SDP	39
3.6	Agentes em IAD	41
3.6.1	Em direção a um conceito de agente	41
3.6.2	Um Modelo de Agente	42
3.6.3	Agentes Cognitivos (AgC's)	44
3.6.4	Agentes Reativos (AgR's)	44
3.7	Comunicação entre Agentes	46
3.7.1	Tipos de Comunicação entre Agentes	46
3.7.2	A Comunicação sob a Ótica deste Trabalho	48
3.7.3	Elementos e Conceitos	49
4	Lógica Nebulosa	54
4.1	Introdução	54
4.2	Resumo Histórico	54
4.3	A Lógica Nebulosa	56
4.3.1	Teoria dos Conjuntos Nebulosos	58
4.4	Operadores da Lógica Nebulosa	59
4.4.1	Operador t-norma (\circ):	60
4.4.2	Operador s-norma ou t-conorma (\square):	60
4.4.3	Operadores Variantes	60
4.5	Conhecimento Impreciso	62
4.6	Regras de Produção Nebulosa	63
4.7	Cálculo de Inferência	65
4.7.1	1. Passo: “Fuzzificação”	65
4.7.2	2. Passo: “Matching” ou Casamento	66
4.7.3	3. Passo: Mediação	66
4.7.4	4. Passo: “Desfuzzificação”	67
4.8	Alguns Comentários	67
5	Uma Arquitetura para Agentes Autônomos	68
5.1	Introdução	68
5.2	Arquitetura de Base	68
5.2.1	Os Três Níveis da Arquitetura	69
5.2.2	Fundamentação Biológica da Arquitetura	69
5.2.3	Considerações sobre a Arquitetura Proposta	72
5.3	Apresentação dos Protótipos	73
5.3.1	Brooks	73
5.3.2	Steels	74
5.3.3	Maja	74
5.4	Modelo Físico do Agente Autônomo	74
5.5	Nível Reativo	79
5.5.1	Conexões Neurais	79
5.5.2	Agentes Nebulosos	81
5.5.3	A Semântica dos Agentes	82

5.5.4	Outro Circuito	84
5.5.5	Estendendo os Circuitos	86
5.5.6	Algumas Considerações	89
5.6	Nível Instintivo	90
5.6.1	Agentes Instintivos	91
5.6.2	Seletores de Agentes Instintivos	98
5.7	Nível Cognitivo	103
5.7.1	Brooks	104
5.7.2	Steels	106
5.7.3	Maja	107
5.8	A Arquitetura Completa	111
5.8.1	Observações Finais	113
6	Implementações e Resultados	114
6.1	Introdução	114
6.2	Considerações Iniciais	114
6.2.1	A Implementação	114
6.2.2	Coordenadas de Navegação	115
6.2.3	Amarração dos Pontos	117
6.2.4	Cálculo da Rotação	119
6.2.5	Cálculo da Interseção dos Cones com Objetos	120
6.3	Protótipo Brooks	121
6.4	Protótipo Steels	123
6.5	Protótipo Maja	126
7	Conclusões	132
7.1	Introdução	132
7.2	Retrospectiva da Tese	132
7.3	Resultados Alcançados	135
7.4	Comparativo com Outros Trabalhos	136
7.4.1	Lógica Nebulosa:	137
7.4.2	Visão Conexionista:	139
7.4.3	Visão da IA Simbólica:	140
7.4.4	Visão de Esquemas:	142
7.4.5	Visões Alternativas à IA:	144
7.4.6	Visão quanto à Aprendizagem:	146
7.4.7	Visão Evolucionista:	147
7.5	Trabalhos Futuros	149
A		153
A.1	Bases de Regras	153

Lista de Figuras

1.1	Principais Paradigmas de Arquiteturas de Controle para AA's	6
1.2	Seqüência dos Capítulos	17
2.1	Planejamento de Locomoção	28
3.1	Modelo Genérico de Agente	42
3.2	Esquemas de cooperação e competição	50
3.3	Outros conceitos em IAD	52
4.1	Descrição da variável salário	63
4.2	Curva granular para duas variáveis V_1 (antecedente) e V_2 (conseqüente)	65
5.1	Visão Geral dos Três Níveis dos AA	70
5.2	Sonares Locais do Agente Autônomo	75
5.3	Circuito Neuronal - Unidirecional	80
5.4	Circuito de Agentes - Bidirecional	84
5.5	Circuito de Agentes (Controle à esquerda, à direita e à frente)	86
5.6	Circuito de Agentes (Controle à esquerda, à direita e à frente)	87
5.7	Circuito de Agentes (Controle à esquerda, à direita e à frente)	88
5.8	Circuito de Agentes (Controle à esquerda, à direita e à frente)	89
5.9	Rede Neural Equivalente ao Conjunto de Regras de Produção	101
5.10	Curva Triangular de Pertinência	102
5.11	Nível Cognitivo - Brooks	105
5.12	Nível Cognitivo - Steels	107
5.13	Nível Cognitivo - Maja	109
5.14	Arquitetura Completa	112
6.1	Mapa gerado por ângulos e distâncias	116
6.2	Vértices dos cones do AA	117
6.3	Detalhe da metade de um cone sensorial	118
6.4	Navegação em ambiente "lunar"	121
6.5	Ambiente típico de um labirinto	122
6.6	Ambiente típico do centro da cidade	123
6.7	"Bandeira do Brasil"	124
6.8	Ambiente supermercado	125

6.9	Ambiente com obstáculos frontais	126
6.10	Ambiente com obstáculos frontais	126
6.11	Ambiente de um beco com uma fonte atratora	127
6.12	O agente visita outras marcas	128
6.13	Saída tendenciosa pela esquerda	129
6.14	Aleatoriedade na escolha esquerda ou direita	130
6.15	Sala quase fechada	130
6.16	Tempo maior no estado de caminhar aleatoriamente	131
7.1	Comportamento Itinerante com Orientação	150

Lista de Tabelas

3.1	Temas de Pesquisas (SDP,SMA) e Propriedades	40
4.1	Operadores Duais t-norma e t-conorma	61
4.2	Principais Operadores de Implicação	61
4.3	Curvas dos Termos da Variável Salário	63

Capítulo 1

Introdução

1.1 Inteligência Artificial

Um objetivo recorrente em Inteligência Artificial (IA) é a especificação de metodologias e técnicas que resolvam problemas específicos, cujas soluções apresentem características do comportamento humano “inteligente”. Os objetivos atuais da IA estão distantes da definição original de inteligência para uma máquina, formulada por Turing, que exige senso comum em sua avaliação, e é contestada por alguns autores[Bro91d]. A IA atual visa estabelecer uma tecnologia capaz de suportar o desenvolvimento de programas com bom desempenho em tarefas que exijam sofisticação cognitiva em um domínio especializado. Um programa que atenda a essa condição é considerado inteligente. Essa nova definição de “inteligência” almeja um sistema com a capacidade de aprender novos conhecimentos a partir de um conhecimento básico e, finalmente, reproduzi-los com desempenho semelhante a um especialista. Exemplifica-se tal definição com os sistemas especialistas de capacidade de aprendizado.

Seguindo essa definição preliminar de “inteligência”, as pesquisas de IA guiaram-se por abordagens *incrementalistas* [Bro91c], desenvolvidas sob a *hipótese simbólica* [Bro91b], de modo que haja uma equivalência plausível entre o computador digital e a capacidade de apresentar um nível razoável de inteligência. Com isso, os sistemas construídos a partir de modelos eram dirigidos a uma manipulação simbólica. Os modelos destinavam-se a tornar o mundo real tratável, utilizando-se de símbolos. A hipótese era validar que o incrementalismo do conhecimento conduzisse a sistemas mais inteligentes.

A IA clássica tem trabalhado sob uma visão *analítica*, na qual o conhecimento é concebido como uma hierarquia de subsistemas, que fundamenta uma *representação de conhecimento*. Nesse sentido, há uma analiticidade, que procura a separação do inteiro em partes individuais menores [Lan92]. Após essa divisão, o sistema é estudado e modelado, de modo que se obtenha representações fiéis das partes, visando o estudo do sistema inteiro.

Um objetivo imediato da IA nessa abordagem analítica impõe a busca por *rep-*

representações de conhecimento eficientes. A área de Representação do Conhecimento (RC) tem tido importância fundamental na IA nestas últimas décadas. A RC aborda os métodos que retratam de forma sistemática os elementos de um sistema. A RC visa métodos que permitam construir modelos computacionais consistentes para que os caminhos entre os mundos físico e simbólico sejam únicos nos dois sentidos. Essa abordagem tem sido aplicada com sucesso para os sistemas chamados de *assistentes humanos* [Bro91c]. Exemplos desses sistemas são: sistemas especialistas (SE's), provadores de teoremas, sistemas de buscas, construtores de SE's, etc.

Contudo, a abordagem analítica não parece adequada para uma certa classe de problemas, representada por sistemas, cuja redutibilidade é praticamente impossível. Esses problemas referem-se a sistemas que apresentam uma alta complexidade em seus níveis internos, um comportamento *não-linear*, um forte dinamismo com o ambiente e uma evolução ao longo de seu processo de existência. O fator sensitivo desses sistemas reside no acoplamento direto entre seus campos perceptual e motor. Outro ponto é a natureza distribuída e paralela de seus processos. Nesses casos, a não-linearidade é perdida quando tais sistemas são tratados isoladamente. De modo resultante, esses sistemas exibem uma boa capacidade de adaptação em face às novas situações ambientais, que são típicas de seres biológicos [Lan92]. Tais sistemas são capazes de se adaptar de forma mutável a uma situação. De imediato, pode-se pensar que essa classe de problemas não pertença à IA clássica. Mas um dos objetivos iniciais da IA não era estudar a modelagem do conhecimento humano a fim de representá-lo de forma abstrata?

Dos questionamentos originais da comunidade de IA sobre a reprodução da inteligência humana, esse objetivo tem se reduzido a subproblemas especializados. Exemplifica-se com problemas como: compreensão da linguagem natural, visão, voz, representação do conhecimento, verificação de planos, etc. São tarefas delimitadas em suas áreas. Segundo R.A. Brooks [Bro91c], *“há um grupo de sonhadores da IA (não sonhando com dinheiro), que imaginam que algum dia essas peças todas cairão em seus lugares apropriados e que sistemas verdadeiramente inteligentes vão surgir”*. Imagina-se uma inteligência emergente, fruto da aglutinação de diversas soluções particulares, afastando-se com isso de propostas universais para inteligência humana, como o exemplo do GPS (“General Problem Solving”), idealizado na década de 60 por A. Newel e H.A. Simon [Win87].

R.A. Brooks [Bro91c] argumenta dois pontos sobre a questão acima:

- A complexidade e a pouca compreensão a respeito da inteligência humana dificulta o procedimento de dividi-la corretamente em “pedaços” com interfaces consistentes;
- A necessidade de práticas com inteligências primárias. Essas práticas buscam traduzir experimentos físicos, enfocando uma modelagem biológica de animais primários [AD91, RDBS91].

Esse último procedimento ocorre por sucessivos ciclos na construção e experimentação da inteligência em animais inferiores. Contudo, as dificuldades históricas

levantadas pela IA ainda persistem: o problema da construção de uma máquina (hardware e software), que simule a inteligência humana a um nível razoável de aceitação, permanece ainda insolúvel¹.

Atualmente há um objetivo comum por parte das comunidades de engenheiros e biólogos em construir sistemas artificiais baseados em hipóteses e suposições de uma inteligência mais próxima à natureza biológica. Todos visam uma melhor compreensão do comportamento inteligente humano no que concerne à sua individualidade, coletividade, capacidade de adaptação, . . . , etc, e como tudo isso pode emergir em uma forma *criativa*. As questões envolvem estudos interdisciplinares em Psicologia [Slo94], Sociologia, Etologia, Ecologia, etc. Nesse meio tempo, surgiram outras áreas visando a solução de problemas não abordados ou não resolvidos pela IA clássica. Dentre essas áreas destacam-se: Redes Neurais Artificiais (RNA's), Vida Artificial (VA) [Lan92] e a Inteligência Artificial Distribuída (IAD) [Gas91].

Paralelamente às novas áreas e à IA clássica, há uma área emergente chamada de *sistemas autônomos* (SA's), que objetiva criar novas abordagens para uma "inteligência autônoma". Sistemas que apresentam um elevado grau de autonomia têm um interesse prático e teórico em vários segmentos da ciência. De imediato, revela-se uma analogia entre os SA's e os seres vivos em suas múltiplas facetas. Por exemplo, um programa embarcado em uma nave espacial, frente a uma situação inesperada no espaço, pode necessitar de uma reconfiguração para continuar sua missão. Nesse caso, um programa de interesse deve ser capaz de se *auto-reconfigurar*. Uma abordagem clássica pode ser impraticável nesse caso, pois o número de situações possíveis torna proibitiva a armazenagem de tal conhecimento. Além disso, as distâncias envolvidas tornam inviável a comunicação direta ou ainda pode ocorrer a limitação do tempo no disparo de uma ação. Num segundo exemplo, tem-se uma consulta em uma rede de computadores. A consulta assume o papel de um processo autônomo ou agente destinado a buscar dados em vários nós da rede. O processo apresenta uma independência quanto a um nó de destino, de modo que cada nó é um contexto desconhecido.

A partir dos dois exemplos acima, preliminarmente, introduz-se o termo *agente* para um programa ou um processo. Os agentes possuem uma autonomia no seu ambiente de atuação², isto é, são *agentes autônomos* (AA's) [Mae91a, Ste94b, Ste94a]. Outros exemplos de agentes implementados como programas de computadores podem ser obtidos em [Dem93].

1.2 Agentes Autônomos

Um AA é um sistema formado por um conjunto de elementos com uma particular relação entre si e o ambiente [Ste95b]. Não necessariamente, o agente é objeto de uma instanciação física unificada [Etz93] como um robô móvel, mas ele pode ser

¹Recentemente, Brooks e outros passaram a construir o primeiro humanóide, o COG. Esse projeto é fruto de muitas experiências com robôs de menor escala [Ste94e].

²Esses dois exemplos há programas residentes em CPU's de computadores.

constituído por um programa de computador ou por um conjunto de objetos qualquer, que possuam uma interação permanente ou não com o ambiente. Por exemplo, um programa residente em um sistema de computador que lhe comunique a chegada de uma nova mensagem cuja origem e/ou conteúdo foram previamente especificados também pode ser considerado um agente. Recentemente, no entanto, os AA's fisicamente implementados que desempenham alguma atividade em particular no mundo real têm sido objeto de intensas pesquisas [HR95]. Por exemplo, esse tipo de agente exige que se leve em consideração restrições de operações semelhantes aos instintos biológicos de autopreservação ou manutenção de vida.

De acordo com L. Steels [Ste95b], esse instinto de autopreservação se manifesta em diversos níveis da biologia. Steels afirma, ainda, que níveis equivalentes podem ser definidos para agentes robóticos. Sob essa visão caracterizam-se os níveis *genéticos*, *estruturais*, *individuais* e de *grupo*. No nível genético, os agentes estão preocupados com a reprodução e perpetuação da espécie. No nível estrutural, preocupam-se com a construção de componentes e processos individuais relativos aos agentes. No nível individual, retoma-se a questão de um comportamento apropriado num dado ambiente, objetivando uma automanutenção³. A nível de grupo, há um mecanismo social que visa manter os interesses coletivos em primeiro plano, isto é, um objetivo social de sobrevivência em um ecossistema. Por exemplo, em ambientes típicos de colônias sociais (as formigas, as abelhas, as bactérias, etc), o auto-interesse é mínimo ou até mesmo inexistente.

Os níveis descritos por L. Steels, têm como objetivo criar uma nova definição para inteligência, diferente do tradicional teste de Turing. Uma definição mais envolvida com fundamentos biológicos e menos controversa⁴. Nesse sentido, verificam-se algumas lacunas na IA sobre a definição de inteligência, basicamente por dois motivos:

- Ausência de uma conexão mais estreita entre a IA e os mundos físico e biológico;
- Ausência de técnicas dirigidas a estruturas e esquemas [MAA95] para conhecimentos e comportamentos na área sensorial-motora, que acompanhem o ciclo tradicional da IA de analisar um problema, formalizá-lo e programá-lo de forma corpórea⁵;

A área de agentes autônomos inteligentes pretende resolver essas duas questões fundamentais. A meta é vislumbrar teorias que se aproximem das leis naturais da física e da biologia para que uma idéia de inteligência possa emergir a partir de processos elementares. L. Steels [Ste95a] frisa que a área de inteligência sensorial-motora segue duas novas diretrizes que preenchem as questões levantadas:

- Uma orientação biológica no sentido de que inteligência é automanutenção. As chances de sobrevivência implicam em funcionalidades sofisticadas de um AA,

³Leia-se sobrevivência.

⁴Uma outra referência contestando o teste de Turing encontra-se em [Kak96].

⁵Do inglês “*embodiment*”.

como: representação, especialização, cooperação, reflexão, comunicação, etc, em nível perceptual e ambiental. Tudo isso com uma fraca ênfase de níveis cognitivos ou superiores;

- Uma compatibilidade das leis físicas, químicas e biológicas como proposta de uma teoria universal. A inteligência é um resultado de um processo não-linear, que exhibe propriedades encontradas em vários tipos de sistemas. Por exemplo, a auto-organização é encontrada nas reações químicas onde as moléculas procuram se agrupar de acordo com alguma afinidade (contudo, esse evento está ligado a níveis inferiores aos das moléculas).

Certamente, parte dessa inteligência tem como origem a interação dinâmica de estruturas e processos com o ambiente num acoplamento em vários níveis. A partir das interações é que se percebe a complexidade do acoplamento entre o AA e seu ambiente. Devido a essa dificuldade, os pesquisadores enfatizam dois ingredientes em projetos de AA's inteligentes: a *situação* e a *incorporação*. O fato do agente estar situado reflete o nível de seu acoplamento com o ambiente. A corporeidade ou incorporação reflete uma instância física do agente com relação ao ambiente.

De modo restrito, L. Steels propõe que a inteligência, bem como a definição de AA's, estejam associadas à visão dos sistemas biológicos. Os sistemas biológicos possuem uma estrutura interna prévia que não foi construída por uma entidade externa. Sendo assim, a inteligência se desenvolve a partir de esquemas e estruturas internas geradas e mantidas por mecanismos como, a *auto-organização*, a *evolução*, a *adaptação* e a *aprendizagem*. Esse processo permanece enquanto for viável a operação em seu ambiente. O conceito é fortemente ligado à automanutenção com uma auto-regulagem, considerando suas próprias motivações. Apesar de tais aspectos tão desejáveis a sistemas autônomos, atualmente, ainda não exibem tais potencialidades [Ste95b]. Tais sistemas podem não existir mas apontam direções de muitas pesquisas.

Em face às mudanças ambientais, a *adaptação* tornou-se um dos critérios em projetos de AA's. Uma das mais importantes características de um ser vivo e ser inteligente é a sua capacidade de adaptação. Os seres vivos, em geral, exibem um alto grau de acomodação em seus comportamentos, apesar de isso não implicar em caracterizá-los como seres inteligentes. O detalhe é que esses seres possuem estruturas físicas internas, que lhes propiciam que processos como a aprendizagem e a adaptação ocorram numa velocidade espantosa se compararmos a qualquer programa computacional [Bro92]. Isso leva a crer que adaptação e inteligência são conceitos que se complementam de alguma forma. Nesse ponto, a inteligência é vista em múltiplos níveis internos com diversos acoplamentos ambientais, bem como um relacionamento parcial interno (apenas por necessidade) entre os diversos níveis. De algum modo, as representações desses níveis são implícitas.

Em resumo, as estruturas físicas exibidas por um ser biológico são análogas às arquiteturas de controle interno dos AA's. Tomando-se por base tais arquiteturas, novos conceitos de adaptação e inteligência são desenvolvidos. Há um ciclo desencadeado a partir de uma estrutura primária no que tange a: sua corporeidade, sua situação nos

diversos níveis com o ambiente, sua adaptação e uma possível inteligência emergente. Nesse ciclo é que arquiteturas primárias são avaliadas como possíveis estruturas para controle de AA's. Em seguida, uma sinopse é apresentada dos principais paradigmas das arquiteturas de controle para AA's.

1.3 Arquiteturas de AA's

Num enfoque situacional, Maja Mataric [Mat94] apresenta uma proposta de classificar os paradigmas das arquiteturas de controle para AA's. O enfoque foge da IAD, mas situa-se nessa área debutante de sistemas autônomos [Ste95b]. O aspecto do controle é apenas uma característica das arquiteturas, levando-as a serem adaptativas ou não. As arquiteturas propostas e implementadas possuem um controle que segue desde um paradigma totalmente *simbólico* a um *puramente reativo*. A classificação é ilustrada pela figura 1.1. Essas arquiteturas apresentam como tendência em suas implementações características como: pouco uso de técnicas matemáticas, algumas idéias elementares da IA clássica e fundamentos de outras áreas emergentes (redes neurais, algoritmos genéticos, lógica nebulosa, etc).

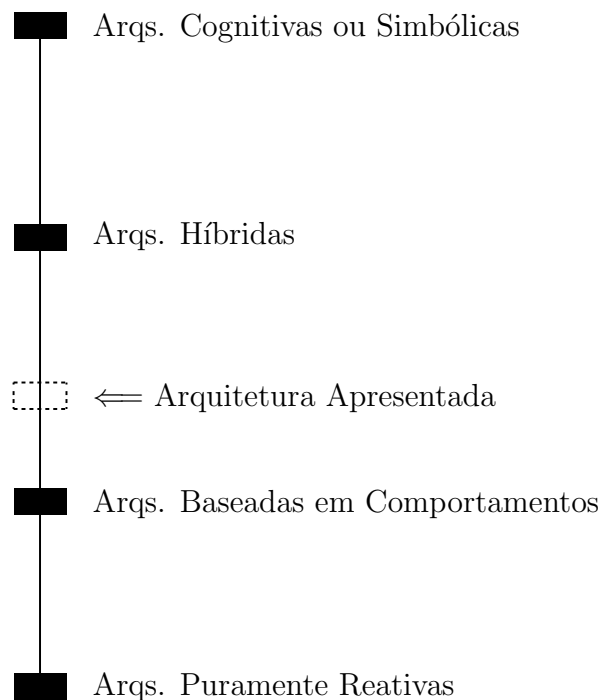


Figura 1.1: Principais Paradigmas de Arquiteturas de Controle para AA's

Em um dos extremos, tem-se as abordagens tradicionais com estratégias de controle *baseadas em planejadores*, que empregam um modelo de mundo centralizado na verificação das entradas sensoriais e na geração de ações de acordo com o contexto. A

informação extraída do modelo de mundo é empregada para produzir uma sequência de ações ajustadas ao contexto, no caso um plano. Há um modelo para mundo real, onde um controlador centralizado segue uma busca inferindo a exploração de estados. Um exemplo dessa abordagem baseada em planejadores é encontrado no capítulo 2 da tese [RK94, Lau90]. Vale salientar que esse tipo de arquitetura exhibe aspectos mais robustos quanto à correteza de seus métodos e análises gerais de um procedimento selecionado.

Em um outro extremo, tem-se as abordagens *puramente reativas*, que implementam uma estratégia de controle como uma coleção de pares de condição-ação. São sistemas que consistem em uma *coleção de regras puramente reativas*, contendo os estados mínimos internamente. Num paralelo a seres biológicos, essas regras refletem os instintos sensoriais básicos dos animais. Por exemplo, qual a reação de um pequeno animal diante do seu principal predador? Esses sistemas não apresentam nenhum modelo interno e nem realizam nenhum tipo de pesquisa. Meramente, selecionam e comandam as ações mais apropriadas para cada conjunto de leitura dos sensores. Há um acoplamento direto entre sensores e a ação, em uma realimentação rápida e dinâmica com o ambiente. O exemplo mais expressivo das arquiteturas reativas é o *englobamento* ou *subjugação*⁶ [Bro91c, Bro91b, Bro91d, dS96].

As arquiteturas *híbridas* apresentam um compromisso entre as duas abordagens citadas acima. A parte reativa opera em baixo nível, enquanto que um planejador toma as decisões de alto nível. O controle é separado em dois ou mais módulos com uma comunicação independente entre eles. De um modo geral, a parte reativa se preocupa com a segurança do sistema (exemplo: permissão em tocar nas paredes, afastar-se de locais perigosos, etc), enquanto que um planejador seleciona seqüências de ações. Segundo [Mat94], essa abordagem é também considerada do tipo baseada em planos, uma vez que há um controlador centralizado de objetivos.

As abordagens baseadas em *comportamentos* (ver figura 1.1) seguem um estilo próximo aos princípios das arquiteturas reativas [Mat92]. Nesse caso, há um mapeamento de condição-ação, embora não se considere um projeto “completo” de agente. Elas apresentam muitas características dos sistemas reativos, contudo não se limitam a uma seleção-ação a cada instante. Os sistemas baseados em comportamentos possuem diferentes formas de representação interna e desempenham algum tipo de computação para tomar uma decisão para uma ação contextual. Como propriedades comuns com os sistemas reativos tem-se a natureza distribuída e concorrente. Contudo, o foco é um comportamento emergente característico devido à ausência de um esquema de raciocínio centralizado. Há regras que definem comportamentos mais elaborados que as puramente reativas. Sumarizando-se algumas características dessa abordagem e em certo grau também das arquiteturas reativas, tem-se:

- A inexistência de um controlador central para o conjunto de todas entradas do sistema. As entradas são fundidas em um módulo comportamental segundo um conjunto de ações de interesse;

⁶Do inglês “*subsumption*”.

- A existência de comportamentos ordenados segundo uma hierarquia de interesses. Esse item não se aplica a arquiteturas reativas, uma vez que não admitem uma coordenação de partes;
- A construção é tipicamente “*bottom-up*”. Esse detalhe difere de todos os sistemas de IA em que a cognição é modelada segundo uma arquitetura “*top-down*” [Bro91b, Mat92];
- Há uma busca em maximizar as informações diretas com o mundo e minimizar a comunicação interna entre as partes do sistema. Nos sistemas reativos há muito mais ligações entre os módulos;
- Há uma homogeneidade entre os módulos comportamentais sem que exista distinção entre o que é reativo e o que é cognitivo.

Uma análise comparativa dessas quatro abordagens quanto ao poder de resolução de problemas é necessariamente complexa e controvertida [Mat94]. Aparentemente, um sistema reativo é menos poderoso que um sistema baseado em planos. Contudo, isso pode não ser verdade se um agente puder ser modelado em condições de contorno parciais do mundo real, isto é, com todos estados relevantes⁷ que o sistema puder encontrar e com ações eficazes correspondentes. Considerando o avanço da tecnologia de sensores e o fato da computação ser mínima em sistemas reativos, eles são mais eficientes que os baseados em planos na maioria dos casos. Devido a essa eficiência e aos outros argumentos citados acima, os sistemas reativos têm sido objeto de diversas pesquisas e representam uma tendência a ser seguida.

Embora a área de AA's tenha uma grande relação com a área da robótica (que trata sobre movimentação de robôs, sincronização, graus de liberdade), existem diferenças básicas entre robôs tradicionais e AA's:

1. Na autonomia propriamente dita, os robôs sugerem o uso de bases operacionais (“on-line”) para uma interação externa. As bases operacionais correspondem a sistemas externos de controle que definem uma seqüência de ações ao robô;
2. Na abordagem *baseada em conhecimento explícito* que a robótica realiza no seu processo inferencial, “...há um processo de raciocínio sobre um modelo simbólico interno” [Mae91b]. Os AA's propõem-se a ter uma arquitetura distribuída, de modo que inferências não ocorram sobre um modelo centralizado de conhecimento;
3. Na ênfase das arquiteturas dos AA's em que há um acoplamento direto entre a percepção externa e a ação motora. Os AA's exibem uma característica bastante *reativa* [DM91], visando uma operacionalidade em tempo real;

⁷Leia-se significativos ou característicos.

4. Na distributividade e descentralização com que as ações são inferidas pelo agente, isto é, as arquiteturas reativas em AA's, a ação emergente ocorre pela fusão dos processamentos resultantes nos vários módulos de comportamentos elementares;
5. Na presença em AA's de mecanismos intrínsecos, que tratam das limitações de recursos e conhecimento incompleto.

As diferenças mencionadas acima ressaltam alguns pontos-chaves e características entre AA's e robôs. Existem algumas idéias e proposições, contudo não há nenhum tipo de consenso no projeto de arquiteturas [MB94]. Numa primeira instância buscase analogias com a biologia, aproveitando o seu processo evolutivo de milhões de anos.

O atrativo de projetos de AA's encontra-se em suas numerosas aplicações práticas em várias áreas, como robôs, softwares, dispositivos específicos da automação industrial, naves espaciais, ações teleguiadas, etc. Mais recentemente vislumbrou-se a possibilidade de vida extraterrestre. Portanto, há necessidade de se realizar viagens a outros planetas. Essas viagens interplanetárias podem durar meses ou anos. Para isso, o envio de seres humanos é ainda um problema de custos e riscos elevados. Como alternativa enviam-se robôs exploratórios, preferencialmente, uma legião desses. O ganho da missão ocorre pelo paralelismo dos dados obtidos, pela distribuição dos AA's em várias regiões e pela robustez da missão quanto à perda ou à pane de um agente. Esse tema é explorado em [Fly89, Bro90], onde minúsculos AA's são especificados para uma missão planetária.

Para sumarizar a discussão acima, apresentam-se algumas características gerais desejáveis na implementação de um AA [Mae91b, Mae91c, Ste91]:

- **Funcionalidade Emergente:** *“a funcionalidade de um agente é vista como uma propriedade emergente a partir de uma interação intensa do sistema com o seu ambiente dinâmico. A especificação do comportamento isolado do agente não explica a sua funcionalidade, que é apresentada quando esse se encontra em operação”* [Mae91b]. Em vez disso, a funcionalidade é fundamentada nas propriedades ambientais e nas suas possibilidades exploratórias. Como uma importante implicação, tem-se o fato de que o objetivo de um AA muitas vezes não é definido. Com um laço interativo entre sistema e ambiente ocorre uma convergência em torno de um objetivo desejado;
 - **Decomposição a nível de tarefas:** *“Distributividade e implementações paralelas são vistas como ingredientes cruciais para comportamentos robustos e rápidos”* [Mae91b]. Os AA's não apresentam um modelo global interno, de modo que não há uma atividade global de planejamento com uma estrutura hierárquica de objetivos. O objetivo tende a emergir da interação interna entre os módulos do AA e/ou com o ambiente. Fundamentalmente, as tarefas são decompostas e associadas aos diversos níveis internos do AA. Os níveis refletem os acoplamentos sensório-motor com o ambiente;
-

- **Métodos de raciocínio que manipulam representações próximas a dados sensoriais:** “*Isto deve-se ao trabalho da percepção que é menos geral e mais realístico*” [Mae91b]. Segundo L. Steels [Ste91], há um enorme esforço na extração de dados e construção de modelos simbólicos para o raciocínio de uma ação. Normalmente, os modelos simbólicos, típicos da IA, são muito complexos e de alto custo computacional numa operação⁸ em tempo real. Além disso, esses modelos simbólicos muitas vezes não são transparentes o suficiente para assegurar uma boa relação entre as entradas e as saídas. Contudo, a ausência completa de um modelo interno em AA’s apresenta severas limitações [Ste91]. A principal restrição é a dificuldade de convergência em torno de uma funcionalidade interessante. Nesse caso, um conhecimento cognitivo simbólico pode provocar em determinados contextos um aumento significativo na competência de AA’s puramente reativos.

Devido ao pouco tempo de existência na linha de pesquisa de AA’s, aproximadamente 10 anos, existem diversas perguntas não respondidas no que diz respeito a uma proposta de arquitetura [Bro91c], a um paradigma de implementação, a experimentos sobre evolução de vida artificial [Pra94], à síntese de comportamentos, etc. A área de AA’s é emergente quanto às suas pesquisas, multidisciplinar e sujeita a novas abordagens, tratando-se das questões de suas possíveis arquiteturas.

1.4 A Motivação do Tema

Nesta seção apresentam-se algumas razões que motivaram o desenvolvimento desta tese. As razões são vistas sob diferentes ângulos. Primeiramente, o termo “*arquitetura*” é discutido em um contexto da IA. Segundo, discute-se qual arquitetura usar em AA’s, principalmente na questão da locomoção. Terceiro, por que desenvolver AA’s com bases biológicas. E, finalmente, por que investir na difícil questão de construir um modelo completo de um AA com a habilidade da locomoção.

1.4.1 Arquiteturas em IA

A IA, bem como as Ciências Cognitivas, têm dirigido suas pesquisas sobre domínios específicos como: visão, reconhecimento de voz, SE’s, mecanismos e formas de representação de conhecimento, tratamento da informação, lógicas inferenciais, etc. São componentes isolados que possuem funções particulares. Mas a procura de modelos que integrem esses componentes e propiciem interações tem sido um tema recorrente dentro da IA, seja uma implementação simulada ou um experimento físico [Ste94e]. Observa-se uma constante busca de modelos sofisticados e completos em suas funcionalidades. Mas para isso, necessita-se que tais habilidades específicas se inter-relacionem, o que é possível usando-se uma *arquitetura*.

⁸Construção e manutenção de um conhecimento.

As arquiteturas surgem como respostas ao problema de combinar essas habilidades de forma coerente, em diferentes níveis de abstração, visando funcionalidades diversas. Segundo A. Sloman [Slo96b], “*arquitetura*” é um tema relacionado a tarefa de definir uma ontologia para *objetos mentais*, *estados* e *processos* (percepções, crenças, desejos, inferências, intenções, humor, aprendizagem, ações, etc). A exploração de uma arquitetura pode revelar características inesperadas sobre a ontologia suportada, que pode se realimentar com novas idéias sobre ontologias e requisitos de projeto. Há um ciclo implícito em torno de um processo de teorização, projeto, implementação e experimentação. Esse ciclo propõe uma avaliação constante sobre a arquitetura e como ela pode se realimentar numa descoberta de novos princípios, visando ajustes ontológicos e de projeto. Esse ciclo conduziu boa parte do corrente trabalho, convergindo em uma arquitetura específica, que suporta algumas funcionalidades ligadas à locomoção.

Contudo, a capacidade de integração dos diversos componentes funcionais exibida pelas arquiteturas encontra-se distante de ser sistemática por algumas razões:

- As particularidades das tecnologias podem ser inerentemente impróprias a um acoplamento. Por exemplo, uma rede neuronal tem como dados de entrada uma matriz de pixels para uma imagem, que dificilmente será usada por um sistema de regras de produção. Uma transformação equivalente se faz necessária;
- A falta de experimentos completos. Em geral, as integrações são parciais ou incompletas em um ciclo de desenvolvimento. Há uma carência em investigações sobre a integração entre componentes, acarretando em uma análise incompleta do sistema. Por exemplo, um paradigma não implementado onde os argumentos não se relacionam;

Por outro lado, a IAD surge como área alternativa, como elo de novas habilidades⁹ ou componentes isolados. A idéia básica da IAD é que as entidades cooperem de forma produtiva¹⁰. Caso uma das entidades tenha a sua competência aumentada, todo sistema torna-se sensível a essa mudança. Eventualmente, por alguma razão interna, a nova competência pode entrar em desacordo com o(s) objetivo(s) do sistema global. Nesse ponto, há uma sutil diferença entre projetos de IAD e AA's. Em AA's o aumento de competência ocorre pela presença de uma nova entidade. Portanto, o conflito se manifesta de forma incremental e não internalizado.

Unificando a definição de AA's e a solução de um problema, uma arquitetura tem por objetivo relacionar elementos de forma cooperativa e funcional. Há a necessidade intrínseca de uma arquitetura que interligue seus elementos, de modo a permitir a resolução de um problema de maior complexidade.

Em suma, as arquiteturas traduzem funcionalidades pertinentes a projetos de sistemas inteligentes. L. Steels [Ste94b] encaminha esses projetos sob a visão de uma análise comportamental e uma metodologia de construção. O comportamento revela

⁹Leia-se, também, competências.

¹⁰Alguns resultados bem-sucedidos da IAD são encontrados em [Cd95].

a regularidade observada na interação dinâmica entre sistema e ambiente, bem como os processos a eles relacionados. A metodologia de construção trata das abordagens ao problema: a de *modelos computacionais* (equações simbólicas, algoritmos, etc) e *implementações físicas* (a corporeidade). A maioria das aplicações em IA foram conduzidas com *modelos computacionais*. Por outro lado, os AA's procuram ser desenvolvidos segundo uma *incorporação física*.

1.4.2 Locomoção

As aplicações em IA são potencialmente infinitas. Por sua vez, os AA's seguem essa mesma linha. Contudo, a classe de problemas que contrariou as abordagens tradicionais da IA refere-se aos *agentes móveis* [Bro91a]. Os agentes móveis, desenvolvidos por R. Brooks [Bro91d], tornaram-se populares em razão de abordarem um novo conceito de inteligência. Uma inteligência sem representações simbólicas, sem raciocínios sobre modelos . . . , mas emergente a partir de simples elementos conectados entre si e uma forte interação com o ambiente. Os agentes móveis serviram como ferramentas na investigação de um novo conceito para inteligência.

Os AA'ss exploram a dinâmica entre os seus estados internos e o ambiente, sendo que a movimentação de um agente é um de seus comportamentos mais elementares. Assim, as movimentações são analisadas sob os aspectos de suas origens, razões motivadoras, estados mentais, etc, conduzindo a novas experimentações de inteligência.

A locomoção é um comportamento primário comum à maioria dos seres vivos. Tal fato é exemplificado por H.A. Simon [Bro91d], que analisa a caminhada da formiga em uma praia. A sua locomoção é uma habilidade reflexiva e adaptativa em seus mecanismos internos, que seguramente são mais simples que o complexo ambiente da praia. Simon especula que essa analogia possa ser feita a seres humanos [Bro91d]. Com isso, revela-se interesse em se investigar mecanismos simples, no caso a locomoção, como um meio de se entender melhor processos mais sofisticados de inteligência.

Em resumo, a simples locomoção permite avaliar a dinâmica de interação entre AA's e o mundo, promovendo um melhor entendimento de vários conceitos, dentre eles a busca de uma nova avaliação de inteligência. O objetivo é compreender melhor os processos ligados à locomoção, uma das principais funcionalidades que devem ser exibidas pelos AA's [Ste95b, Ste95a], a fim de levantar questões mais elaboradas a respeito da inteligência. Finalmente, um outro ingrediente dirigido a arquiteturas de AA's é a forte inspiração na biologia [Ste94b]. Essas motivações biológicas são discutidas a seguir.

1.4.3 Biologia

Há algum tempo, o ser humano vem se inspirando na natureza de seres biológicos e sua ecologia com o objetivo de desenvolver suas tecnologias. Ele observa um ser vivo que gastou milhões de anos para atingir o estado atual, empresta a idéia, adapta-a ao seu contexto, e utiliza-a para o seu melhor conforto de vida. Realiza um aprendizado geralmente por observação, aplicando-o sucessivamente numa criação típica

do “*homo-sapiens*”. Sendo assim, atualmente, constroem-se artefatos tecnológicos com inspirações em modelos biológicos, em elevado número e graus de aproximação entre a fisiologia animal e sua cópia.

Por exemplo, o movimento das asas dos pássaros e de animais voadores não apresenta uma inspiração prática para os aviões de hoje em dia. Mas a construção de aviões, os detalhes das asas, bicos e seu enfoque aerodinâmico foram fortemente influenciados pelos pássaros e outros animais voadores. Seguindo-se essa linha, áreas como a robótica têm se utilizado largamente dos exemplos da natureza animal. Assim, os robôs restritos em suas implementações físicas deveriam apresentar aptidões próximas aos seres vivos. Outras oportunidades da robótica em reusar o caráter biológico são: geração de ritmos locomotores de pés, manutenção do centro de gravidade estável quando submetido a diversas velocidades e tipos de trajetórias, ajustes compensatórios em terrenos acidentados, etc. Contudo, existem habilidades mais sofisticadas exibidas pelos seres vivos. Em seu habitat natural, eles apresentam processos migratórios quando novos ambientes são apresentados, necessitando adaptar os seus mecanismos de captura de presas num modo temporal, evitando novos tipos de predadores, etc. Enfim, estão sujeitos continuamente a variações ambientais e a novas perspectivas de sobrevivência [RM93].

Num segundo momento, há critérios quantitativos das espécies e suas aptidões ambientais. Levando em conta que há no mundo milhões de espécies de seres vivos, cada qual maximizando o seu tempo de sobrevivência, desenvolveram mecanismos especializados, que normalmente são apropriados e interessantes para serem usados pelo homem. Muitas vezes, um modelo biológico de seres inferiores é mais adaptável que os modelos das vocações naturais do homem. O enfoque sobre esses esquemas e estruturas variam de acordo com interesses do *roboticista* ao *biologista*. Enquanto neuroetologistas¹¹ se preocupam com a configuração neuronal, seus mecanismos de controle e sua relação com o comportamento animal, um biólogo pode enfatizar a química neuronal no processo sináptico. Por outro lado, o roboticista visa a utilização desses esquemas neuronais na construção de mecanismos de controle adaptativo. Todos enfoques e interesses na compreensão da natureza animal visam um avanço de tecnologia. A neuroetologia e as modelagens computacionais ainda se encontram distantes de uma teoria única e linear. Os estudos ainda se mostram sob a forma de pontos quase que isolados, apesar de interseções existirem e continuarem a crescer.

Enfim, constata-se um relacionamento bidirecional entre biólogos e cientistas, na construção de modelos computacionais inspirados biologicamente. Há um arcabouço de conhecimento da biologia e das ciências correlatas, que de modo oportuno pode ser reusado eficazmente. Portanto, há um oportunismo da área de arquiteturas para AA's em usufruir de tais conhecimentos.

¹¹Etologia é a ciência que estuda o comportamento animal.

1.4.4 Modelo Completo

Uma vez delineada a necessidade motivadora em construir arquiteturas para AA's, vislumbra-se uma implementação dirigida à funcionalidade da locomoção¹² sob uma base biológica. Os AA's exibem lacunas nas propostas de arquiteturas e na metodologia para se construir protótipos, procurando-se validar algumas das hipóteses motivadoras.

Basicamente, o artifício utilizado é o da simulação computacional em face dos custos e disponibilidades. O método de validação de uma teoria utilizando-se da simulação, no caso uma arquitetura, é alternativo à experimentação física. A abordagem via simulação permite a exploração e testes de mecanismos relacionados à locomoção ainda não validados na prática. A estratégia permite investigar isoladamente os vários mecanismos impostos por uma arquitetura [RDBS91, Ste94b].

Obviamente que esse procedimento leva a um questionamento do tipo: “*Quanto uma simulação está próxima de um modelo real*”? Uma pergunta simples para uma resposta difícil. Essa pergunta é sumariamente respondida assim: “*Arquiteturas devem ser propostas e simulações traduzem experimentações iniciais*”. Há uma árdua tarefa que precisa ser feita. O ganho da experimentação torna-se recorrente aos conceitos exibidos pelas arquiteturas.

1.5 A Proposta da Tese

Na seção anterior foi levantado um conjunto de hipóteses e argumentos sobre arquiteturas de AA's. Uma vez identificada a ausência de padrões e de um consenso sobre o tema, propõem-se uma arquitetura alternativa dirigida aos AA's.

A arquitetura proposta e apresentada nesta tese situa-se entre as comportamentais e as híbridas (ver figura 1.1), por se observarem os seguintes fatos:

- Ausência de um conceito claro de módulos comportamentais;
- Ausência de um modelo de representação do mundo real;
- Ausência de um planejador e decisor único.

A rigor, essa arquitetura possui agentes com *funcionalidades internalizadas* [Pay91], que controlam pelo conceito de *esquemas* [MAA95, dS96], agentes motores de um nível mais baixo. Sendo a *locomoção* a principal funcionalidade suportada pela arquitetura, esse o tema exploratório sobre a arquitetura proposta. Sob essa plataforma arquitetural, protótipos de AA's foram construídos com funcionalidades específicas a partir da locomoção.

A validação dessa arquitetura de base foi feita com a implementação de três protótipos funcionais. A estratégia de consolidação da arquitetura proposta sugere que ela suporte várias aplicações relacionadas à locomoção. As três aplicações,

¹²É associada a conceitos implícitos de inteligência.

no caso AA's, são distintas e instanciadas sobre um mesmo suporte arquitetural. Os três AA's exploram aspectos específicos de comportamentos inteligentes e da capacidade de adaptação. Algumas características dessa arquitetura são: abordagem multinível, heterogênea, distribuída, adaptativa, definição comportamental no nível superior, uma forte inspiração biológica, um desconhecimento do ambiente, e conceitos de multiagentes. O suporte conceitual é fornecido pela IAD. Contudo, algumas complementações existem de maneira oportuna em relação aos conceitos e elementos apresentados em [Cd94, Cd96].

1.6 Contribuições da Tese

Em face desta tese ser completa quanto ao ciclo de desenvolvimento de uma arquitetura, seguindo da concepção à implementação, as contribuições ocorrem ao longo desse processo. A proposta inicia-se com um esboço preliminar, passando da implementação computacional à avaliação dos resultados obtidos por simulação. Finalmente, uma constante realimentação motivada por essa experimentação simulada permitiu definir os seus limites e suas perspectivas. Desse modo, os principais pontos levantados na tese são resumidos em:

- Uma arquitetura para um agente reativo do tipo móvel que explora o conceito de *esquemas* em vários níveis;
 - Uma arquitetura que apresenta características como: raciocínio descentralizado, ausência de um modelo interno, flexibilidade, robustez, hierarquia, adaptativa, ausência de esquemas de memórias ou armazenamentos secundários, e baixo custo computacional;
 - Uma arquitetura que leva a um AA computacional com os seguintes subsistemas distribuídos: um conjunto perceptual que acompanha o estado dinâmico do ambiente, um conjunto motor que responde robustamente com ações externas e um raciocínio distribuído baseado em esquemas segundo um contexto;
 - Uma arquitetura que é conceitualizada internamente sob a visão da IAD, com uma comunidade de outros agentes que apresentam conceitos de cooperação, competição e conflitos;
 - Uma arquitetura com uma forte inspiração na biologia. A inspiração nessa arquitetura acompanha internamente os sistemas neuronais e externamente a etologia animal;
 - Uma arquitetura implementada computacionalmente que permite a exploração de vários aspectos de projetos sobre AA's e IAD;
-

1.7 O Conteúdo da Tese

Nesta seção é sumarizado o conteúdo de cada capítulo da tese, bem como sugestões de roteiros para sua leitura. Os temas nos capítulos foram selecionados e adaptados segundo uma necessidade de conhecimentos à tese.

1.7.1 Os Capítulos Seguintes

- Capítulo 2: apresentam-se alguns tópicos da IA clássica. Inicia-se pela representação do conhecimento, onde o paradigma de sistemas de regras de produção é explorado. Em seguida, os tipos de raciocínio são apresentados visando sua aplicação a esquemas de regras. Na segunda parte desse capítulo, os planejamentos clássico e reativo são discutidos. Este surgiu como uma necessidade mediadora do uso de planos em AA's;
- Capítulo 3: apresentam-se fundamentos e bases conceituais da Inteligência Artificial Distribuída (IAD). Apesar dos AA's serem atualmente uma área à parte, os conceitos da IAD foram extensivamente utilizados nesta tese. Esse capítulo normaliza e fomenta “*um pano de fundo*” conceitual à tese. Para isso, os temas discutidos seguem por abordagens de problemas, conceitos sobre agentes, tipos, paradigmas, tipos de comunicação e conceitos de interação;
- Capítulo 4: apresenta-se um método de raciocínio aproximado da lógica nebulosa. O método é a inferência, e seus principais resultados são delineados. Essa formalização visa uma modelagem de agentes nebulosos a serem usados em uma das instâncias da tese;
- Capítulo 5: apresenta-se todo o núcleo conceitual da arquitetura proposta. Aqui, as escolhas de projetos são discutidas e especificadas. A apresentação é incremental quanto aos detalhes internos da arquitetura;
- Capítulo 6: apresentam-se considerações sobre a implementação computacional dessa arquitetura. Em seguida, os três protótipos de AA's são submetidos a simulações em diversos ambientes. Os resultados obtidos validam a robustez da arquitetura proposta;
- Capítulo 7: apresenta-se um resumo da tese, bem como resultados alcançados. Essa conclusão segue por uma comparação com outros paradigmas e técnicas de AA's. Finaliza-se com possíveis trabalhos a serem desdobrados a partir desta tese.

1.7.2 Guia para o Leitor

A figura 1.2 sumariza o encadeamento lógico do conteúdo desta tese. Com base nessa sequência de conhecimentos e pré-requisitos, sugerem-se três tipos de leituras para esta tese: uma *rápida*, uma *dirigida* e uma *detalhada*.

- *Rápida*: para o leitor interessado em tomar ciência desta tese e seus resultados. Deve seguir direto ao capítulo 7. Há, ainda, uma panorâmica de contexto de trabalhos e paradigmas relacionados a AA's. Finalmente, alguns pontos em aberto são apresentados;
- *Dirigida*: para o leitor experiente nas áreas de IA e correlatas. Pode realizar um salto e ir direto aos capítulos 5, 6 e 7. No capítulo 5 encontra-se o núcleo conceitual da tese; no capítulo 6, seus resultados a partir da arquitetura desenvolvida e no capítulo 7, a conclusão;
- *Detalhada*: para o leitor interessado em conhecer a área de AA's e a abordagem aqui feita. Sugere-se a leitura dos capítulos em sua ordem natural. As leituras suplementares são encontradas nas referências.

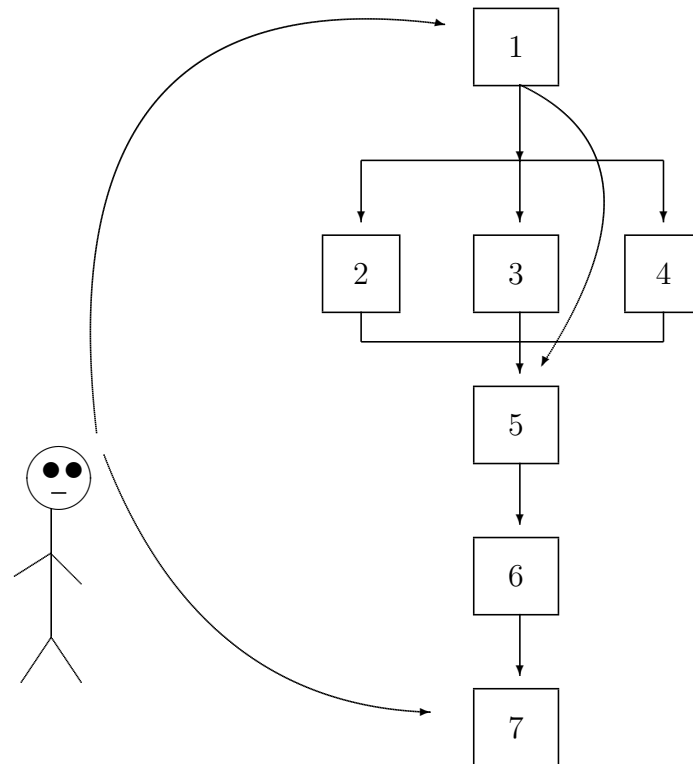


Figura 1.2: Sequência dos Capítulos

Capítulo 2

Elementos de Inteligência Artificial

2.1 Introdução

Neste capítulo, são apresentados alguns temas fundamentais da Inteligência Artificial (IA). A visão sobre os temas aqui discutidos é a *simbolista*. O capítulo divide-se em duas partes principais. A primeira parte apresenta o tema Representação de Conhecimento (RC), e o paradigma abordado são as regras de produção. Parte desta seção foi extraída dos livros de G. Bittencourt [Bit96] e P. H. Winston [Win87]. Numa segunda parte, são apresentadas as idéias sobre *planejamento clássico e reativo*. Dentro do planejamento tradicional, um exemplo é desenvolvido visando mostrar a sua complexidade de representação e métodos de buscas em espaço de estados. O planejamento reativo não se encontra no contexto da IA clássica, contudo é uma linha emergente nas pesquisas em AA's móveis.

2.2 Representação de Conhecimento

A definição do termo Representação de Conhecimento apresenta as mesmas dificuldades filosóficas encontradas para definir o termo Inteligência Artificial. Uma das idéias é definir um mapeamento sintático e semântico de um modelo abstrato ao objeto do mundo real. A representação é um conjunto de convenções sintáticas e semânticas, que visa descrever objetos, situações, ações etc. Contudo, há um consenso de que uma representação deve exibir características como ser: completa, concisa, reveladora de restrições, transparente, computacionalmente eficiente, supridora de pormenores e computável [Win87]. Para se realizar um processo de RC, há três componentes essenciais de uma representação a serem explicitados:

1. o mundo a ser representado (parte física);
 2. o mundo da representação (parte lógica);
 3. a relação entre esses dois mundos.
-

Além desses pontos, o importante é que tanto o mundo a ser representado, como o mundo da representação, devem contar com um conjunto de operadores que permitam a manipulação dos elementos desses mundos. Furbach et al. [FF85] chama o conjunto formado por um desses mundos e seus operadores associados de *Corpo de Conhecimento*.

Resumidamente, a sintaxe de uma representação especifica os símbolos que podem ser usados com suas possíveis construções e arranjos. A semântica de uma representação especifica como e qual o significado que está incorporado nos arranjos permitidos pela sintaxe [Win87]. Exemplifica-se pelas linguagens de programação que representam tipos de procedimentos. A sintaxe, normalmente, é especificada pela forma normal de Backus-Naur. A semântica é descrita pelas construções particulares de elementos da sintaxe, na qual se deseja produzir algo.

O objetivo das representações é definir objetos a partir do mundo real (físico) e estabelecer uma relação com um mundo abstrato ou das representações lógicas, isto é, criar descrições válidas. A relação entre os dois mundos define a *semântica da representação*, informando a relação existente entre os mundos lógico e físico. Essa semântica pode ser expressa de diversas maneiras [Bit96]:

- *Informal*: os significados das expressões utilizadas no mundo da representação são atribuídos intuitivamente a partir dos símbolos utilizados. Esse tipo de semântica é extremamente impreciso e é característico dos primeiros esforços na área da IA.
- *Procedimental*: o significado das expressões do mundo da representação é definido pelo comportamento dos programas que as manipulam. “*Estabelece um conjunto de programas que operam sobre descrições na representação. O significado é definido pelo que os programas fazem*” [Win87]. Esse tipo de semântica também é característico dos primeiros sistemas de inteligência artificial. O fato da semântica de programas ser complexa por si mesma, limita a utilidade dessa abordagem.
- *Descritiva*: o significado das expressões do mundo da representação é estabelecido por descrições precisas, em linguagem natural, em correspondência com o mundo a ser representado. Definem, exatamente, quais são os objetos, as suas ações e os eventos ocorridos nos mundos [Win87].
- *Equivalência*: os significados das expressões do mundo da representação são associados a expressões de uma outra linguagem com semântica estabelecida (lógica, por exemplo).

A tendência atual é o uso de semânticas de equivalências. Nesse caso, abstrai-se o mundo a ser representado, considerando-o como um conjunto de objetos arbitrários, cuja única propriedade é sua identidade. Os operadores que atuam sobre esse mundo abstrato são os operadores usuais da teoria dos conjuntos. O mundo da representação

é definido sob a forma de uma linguagem formal, com operadores definidos sintaticamente. Finalmente, a relação entre os dois mundos é dada por uma função semântica que associa um significado à linguagem de representação em relação à teoria dos conjuntos.

Para evitar as dificuldades filosóficas intrínsecas à definição do termo Representação de Conhecimento, elege-se a seguinte proposta: “*Uma representação de conhecimento é uma combinação de estruturas de dados e de procedimentos de interpretação que, se usados de maneira correta dentro de um programa, levarão a um comportamento que simule o conhecimento dos seres humanos*” [BF81].

Com base nessa definição, a construção e a utilização de sistemas inteligentes apresentam itens relacionados à RC, como: a aquisição de conhecimento, a busca de conhecimento pertinente já armazenado, raciocínios, entendidos como processos geradores de novos conhecimentos a partir do conhecimento disponível, métodos mediadores (consensuais) de raciocínio, interfaces intuitivas com o usuário, etc. Resumindo, todo método de RC, no mínimo, apresenta funcionalidades quanto à *aquisição*, à *busca* e um método de *raciocínio* sobre o conhecimento. De modo implícito, essas são as três etapas do processo de RC.

Como exemplo de RC, optou-se, neste trabalho, pelos *Sistemas de Produção* (SP's). As outras abordagens para esquemas de RC, como redes semânticas, quadros (“*frames*”), lógicas não monotônicas, lógica de 1^a. ordem (predicados), roteiros, dependência conceitual, etc, podem ser encontrados em [Bit96, RK94, Win87]. Quanto à formalização e uso, os SP's são mais antigos e continuam sendo populares no contexto da RC. Qualquer RC deve permitir que vários esquemas de raciocínio possam ser aplicados. O objetivo desse manuseio é a obtenção de novos conhecimentos. Nesses termos, os SP's exibem a característica de suportar vários tipos de raciocínios. Um dos esquemas de raciocínio é o da *lógica fuzzy* [Zad94], apresentada no capítulo seguinte, essa se utiliza do bom formalismo dos SP's para realizar inferências. O método de inferência usado em seu raciocínio é um Modus Ponens modificado. Com base nesses tópicos, tem-se a seqüência das próximas seções.

2.3 Sistemas de Produção

Sistemas de Produção é um nome genérico para todos os Sistemas Baseados em Regras de Produção, isto é, pares de expressões consistindo em uma condição e uma ação. A idéia inicial dos sistemas de produção foi introduzida por Emil Post, em 1943 [Lau90]. Post propôs um conjunto de regras à especificação sintática de transformações sobre cadeias de caracteres, como um método geral para o processamento de dados. Uma montagem axiomática de um *Sistema de Post* é dado por um simples modelo de sistema de produções:

- **Regras:** conjunto ordenado de pares (LE,LD) (LE: lado esquerdo, LD: lado direito da expressão), onde LE e LD são seqüências de símbolos;
 - **Memória de Trabalho** (MT): conjunto não ordenado de símbolos;
-

- **Interpretador:** que executa um procedimento. Uma heurística típica para cada regra (LE,LD) é contínua e executada de acordo com os passos abaixo:

1. Os símbolos de LE estão na memória de trabalho? SIM: substitua os símbolos LE na memória de trabalho pelos símbolos LD.
2. Continue na próxima regra.

Obviamente, há muitas variações em torno de como casar e substituir uma ou mais regras dado um estado corrente na memória de trabalho (MT), alternativas à mudança das regras são do tipo: as que não foram ainda ativadas, as que estão a mais tempo na MT, as que foram usadas recentemente, as com maior grau de compatibilidade, etc. Pois, a princípio, todas as regras são passíveis de serem executadas, uma vez que estão presentes na memória de trabalho.

Em [Win87], há uma ressalva da diferença entre o fato de uma regra estar “*engatilhada*” e ser “*disparada*”. Pode estar engatilhada pelo simples fato de estar na memória de trabalho ou, ainda, ser uma das candidatas selecionáveis. Já o disparo, ele conduz a uma ação que eventualmente muda o conteúdo da memória. Na ocorrência de várias regras serem engatilhadas, há a geração de um conflito da ação: qual a regra a ser disparada? Há necessidade de se estabelecer uma disciplina a ser para se contornar o conflito entre as regras. Em [RK94, Win87], várias estratégias de conflito de regras são sugeridas, contudo, essas devem ser aliadas aos escalonadores de mudanças nas regras da MT.

Exemplificando: à definição do procedimento de RP acima descrito, é dado um conjunto qualquer de regras:

$$\begin{array}{ll} S & \rightarrow ABA \\ A & \rightarrow A1 \\ A & \rightarrow 1 \\ B & \rightarrow B0 \\ B & \rightarrow 0 \end{array}$$

Considerando que inicialmente a memória de trabalho contenha o símbolo S , uma execução do interpretador poderia resultar no seguinte conteúdo para a memória de trabalho:

$$S \rightarrow ABA \rightarrow A1BA \rightarrow 11BA \rightarrow 11B0A \rightarrow 1100A \rightarrow 1100A1 \rightarrow 110011$$

Contudo, muitas outras seqüências poderiam ser obtidas, considerando o método de escalonamento ou substituição das regras na MT. Os SP's foram redescobertos durante os anos setentas como uma ferramenta à modelagem da psicologia humana. O formato *condição-ação* se adapta à modelagem de todos os comportamentos baseados em pares *estímulo-resposta*. Essa idéia é aplicada em muitos momentos deste trabalho. Dois sistemas que utilizaram o modelo de SP's para a modelagem do comportamento humano foram PAS II e VIS [Bit96].

Outros tipos de sistemas que utilizam o formato de regras de produção como método de representação de conhecimento são os Sistemas Especialistas (SE's). O objetivo dos sistemas especialistas é ao mesmo tempo mais restrito e mais ambicioso do que o objetivo dos modelos psicológicos. Os SE's são concebidos para reproduzir o comportamento de especialistas humanos na resolução de problemas do mundo real, mas o domínio desses problemas é altamente restrito. Os primeiros sistemas especialistas que obtiveram sucesso em seu objetivo foram os sistemas DENDRAL e MYCIN [Win87]. O sistema DENDRAL é capaz de inferir a estrutura molecular de compostos desconhecidos a partir de dados espectrais de massa e de resposta magnética nuclear. O sistema MYCIN auxilia médicos na escolha de uma terapia de antibióticos para pacientes de hospital com bacteremia, meningite e cistite infecciosa.

Exemplo de uma regra típica do sistema MYCIN é apresentada a seguir.

SE: o organismo é gran positivo, e a morfologia do organismo é coco, e a formação de crescimento do organismo é aglomerado;

ENTÃO: (0,7) a identidade do organismo é Estafilococo.

Essa regra afirma que na presença de informações confirmando as premissas há uma possibilidade de 70% da identidade do organismo ser estafilococo.

Desde então, muitos SE's foram desenvolvidos para resolver problemas em muitos domínios diferentes, incluindo a Agricultura, Química, Sistemas de Computadores, Eletrônica, Engenharia, Geologia, Gerenciamento de Informações, Direito, Matemática, Medicina, aplicações militares, Física, controle de processos e tecnologia espacial. Alguns desses sistemas, como o sistema XCON/R1 [Win87] para a configuração de computadores VAX da empresa DEC e PROSPECTOR [Win87] para a análise das possibilidades de se encontrar tipos específicos de depósitos minerais em uma dada área, se revelaram altamente lucrativos do ponto de vista comercial.

Com o sucesso dos sistemas especialistas, diversos ambientes de programação especializados no desenvolvimento desse tipo de sistema foram implementados. Esses ambientes, chamados de *arcabouços* ou *construtores* para Sistemas Especialistas, oferecem todos os mecanismos necessários para a implementação de sistemas especialistas, exceto o conhecimento especializado do domínio.

Em geral, a arquitetura de um Sistema de Produção consiste em três módulos: uma *base de regras*, uma *memória de trabalho* e um *motor de inferência*. O motor de inferência controla a atividade do sistema. Essa atividade ocorre em ciclos, consistindo cada um em três fases:

- *Correspondência de Dados* (“*matching*”): se pesquisa quais as regras que satisfazem a descrição da situação atual, para então serem selecionadas;
 - *Resolução de Conflitos*: se escolhe quais as regras que realmente serão executadas, a partir do conjunto que foi selecionado na primeira fase;
 - *Ação*: a execução propriamente dita das regras escolhidas na fase anterior.
-

O teste para verificar se uma regra satisfaz uma dada situação depende de como o conhecimento é representado internamente na base de regras e na memória de trabalho. Esse teste pode ser um processo de unificação entre duas fórmulas lógicas, a comparação entre dois quadros ou simplesmente um teste de igualdade sintática entre dois símbolos. O teste depende também da estratégia de controle adotada pelo motor de inferência. Se o sistema trabalha em um *encadeamento para frente* (“*forward chaining*”), a parte esquerda da regra é posta em correspondência com a descrição da situação que se encontra na memória de trabalho. Esse esquema de raciocínio realiza um processo de *síntese*. Como exemplo, tem-se o sistema XCON [Win87]. Se o sistema trabalha em *encadeamento para trás* (“*backward chaining*”), então a parte da direita das regras é comparada aos objetivos atuais do sistema e suas condições são usadas como novos objetivos caso a comparação tenha sucesso. Já nesse esquema de raciocínio, há um processo de *análise*. Como exemplo, tem-se o sistema MYCIN [Win87]. A sintaxe das regras varia de acordo com o sistema e pode ser bastante convival, tanto no formato de sentenças em linguagem natural, como no sistema ROSIE, ou bastante formais, como na família de sistemas OPS, PROLOG, LISP, etc [Bit96].

As principais vantagens dos sistemas de produção como método de representação de conhecimento são as seguintes: *modularidade*, *uniformidade* e *naturalidade*. As principais desvantagens são: ineficiência em tempo de execução e complexidade do fluxo de controle. O fluxo de controle refere-se ao rastreamento lógico selecionado das regras disparadas. Essas vantagens e desvantagens caracterizam *domínios de uso* que se adaptam no desenvolvimento de SE's baseados em sistemas de produção. Os aspectos relevantes desses domínios são:

- Descrever um conhecimento consistindo em um conjunto muito grande de fatos parcialmente independentes;
- Dispor de métodos de solução, consistindo de ações independentes;
- Apresentar uma nítida separação entre conhecimento e ação.

2.4 Raciocínio

Um processo de raciocínio deve ser capaz de gerar novos elementos, a partir de uma base de conhecimentos previamente armazenados. Uma analogia aos SP's, a presença de um base de regras na MT, estabelece uma condição *sine qua non* a um processo de raciocínio que gere uma nova seqüência de símbolos. Há cinco tipos de processos identificados para o raciocínio: *formal*, *procedimental*, *analógico*, *raciocínio por especialização e generalização* e o *raciocínio em meta-nível*.

O *raciocínio formal* consiste na transformação sintática de elementos de conhecimento pela aplicação de regras inferenciais. Três tipos de raciocínio formal são possíveis:

- Inferência dedutiva: a partir de dois elementos de conhecimento na forma: $\forall x.(P(x) \rightarrow Q(x))$ e $P(a)$, infere-se o novo elemento $Q(a)$. Essa regra de inferência é conhecida como Modus Ponens;

Outras regras de inferência utilizadas no processo dedutivo são:

- Modus Tollens: $\forall x.(P(x) \rightarrow Q(x))$ e $Q(a) \Rightarrow P(a)$;
- Especialização: $\forall x.P(x) \Rightarrow P(a)$;
- Generalização: $P(a) \Rightarrow \exists x.P(x)$;
- Resolução: $P(x)$ ou $Q(a)$, $R(y)$ ou $Q(y) \Rightarrow P(x)$ ou $R(a)$.
- Inferência abdutiva: a partir de dois elementos de conhecimento na forma: $\forall x.P \Rightarrow Q$ e Q , toma-se por hipótese que a razão pela qual Q se verifica é a validade de P , inferindo-se assim uma verdade ao novo elemento P . A abdução é um raciocínio típico aos processos de diagnósticos, mais evidenciado à área médica. Normalmente um médico parte de uma suposta doença, e busca indícios e premissas que a valide ou a refute.
- Inferência indutiva: a partir de fatos experimentais válidos para um P , e uma sucessão desses que levem a uma proposição lógica de um Q , contudo, esse deve ser válido. Desse modo, o elemento Q obtido a partir de: $\forall x.P(x) \Rightarrow Q$; também é válido.

O *raciocínio procedimental* é utilizado sempre que existe um algoritmo para determinar a solução de um problema relevante ao raciocínio corrente. Propriedades geométricas e matemáticas dos elementos de conhecimento são especialmente adequadas para serem inferidas pelo raciocínio procedimental. Inferências associadas a representações analógicas de objetos físicos são freqüentemente procedimentais por natureza. As relações espaciais entre dois corpos, por exemplo, podem ser deduzidas pelo raciocínio formal, a partir de um conjunto de predicados que descrevam suas posições no espaço. No entanto, essas mesmas relações podem ser determinadas muito mais eficientemente, caso se disponha de uma representação analógica do domínio dos objetos e suas coordenadas [Ste91].

O *raciocínio analógico* consiste em determinar relações entre dois domínios, de maneira que métodos de solução adequados para problemas de um domínio possam ser adaptados para solucionar problemas em outro domínio. Embora usual para os seres humanos, apresenta sérias dificuldades para ser formalizado. O mecanismo utilizado pela mente humana para gerar analogias não é bem compreendido e sua aplicação para auxiliar o raciocínio em sistemas de inteligência artificial é bastante limitada [Win87]. Um domínio onde o raciocínio analógico foi aplicado com relativo sucesso foi em demonstração automática de teoremas. A forma restrita das expressões permitidas e a estrutura formal das provas facilitam a definição de medidas de analogia e sua utilização para guiar demonstrações baseadas em outras demonstrações conhecidas.

O *raciocínio por especialização e generalização* consiste na geração de conceitos a partir de exemplos e contra exemplos. Esse método é utilizado principalmente

em programas de aprendizado. Tipicamente, apresenta-se ao programa uma série de exemplos e contra exemplos. O programa deve gerar uma representação em uma linguagem formal com conceitos simples, capaz de explicar as informações disponíveis [Win87]. Esse tipo de raciocínio também é típico de seres humanos e sua utilização em IA apresenta alguns problemas de difícil solução. Embora a disponibilidade de linguagens típicas de IA, como LISP e PROLOG, resolvem a parte formal desse método; a construção automatizada de modelos com essas linguagens são pontos ainda obscuros.

Finalmente, o *raciocínio em meta-nível* consiste na manipulação de informações sobre a utilidade e a disponibilidade do conhecimento sobre o domínio de trabalho. Esse é um importante tipo de raciocínio, pois permite a um sistema inteligente determinar suas próprias limitações e reconhecer problemas que estão além de suas capacidades. Outro uso para o raciocínio em meta-nível é dotar um sistema inteligente com a capacidade de explicar seu próprio raciocínio. Para ser capaz de explicar seu raciocínio, um sistema deve ser capaz de representar e utilizar conhecimento sobre seu mecanismo dedutivo. Essa capacidade é chamada de *introspecção* [RK94].

2.5 Planejamento Clássico e Reativo

As pesquisas na área de IA tradicionalmente visam a uma manipulação simbólica para se atingir um estado ou uma meta desejável. Esse encaminhamento simbolista tem sido focalizado de modo dominante pela área de “*planejamento*”. O planejamento é definido como sendo a geração de uma seqüência de ações destinadas a atingir uma meta. Exemplifica-se o problema sob a visão dos Sistemas Especialistas como sendo uma seqüência de regras de produção (ações), disparadas com objetivo de inferir um resultado. Em resumo, o planejamento é um conjunto de passos possíveis, obtidos de forma sistemática, descrevendo uma trajetória entre um estado inicial e um final.

Mais recentemente, uma linha de pesquisa sobre planejamento considera o aspecto da “*reatividade*” [AC91]. Esse paradigma surgiu na metade dos anos oitentas, de forma a preencher e avaliar melhor as restrições existentes do *planejamento simbólico* para certas aplicações. Essas aplicações apresentam restrições típicas como: tempo de resposta, modelagem pormenorizada do mundo real, desconhecimento, a priori, sob um novo domínio, etc. A ênfase dos sistemas reativos encontra-se na forte interação com o ambiente e no acoplamento “direto” do laço *percepção-ação* na organização de uma atividade. A segunda parte do capítulo apresentam pontos em torno do termo planejamento.

2.5.1 Planejamento Clássico

Basicamente, plano é uma atividade de organizar ações fundamentadas numa construção e execução de estruturas simbólicas, como programas computacionais. Os estudos sobre planos iniciaram-se na década de 50. Uma das primeiras abordagens, feita por Karl Lashley, em 1951, como neurofisiologista, estudava o comportamento

da atividade neural no processo cognitivo da estrutura fonética da linguagem. A idéia era que elementos básicos da linguagem, os fonemas, possuísem estruturas neurais “simples” e não decomponíveis. Uma ordem serial dessas estruturas geraria uma seqüência de ações primitivas pela aplicação de esquemas (“schemata”) habituais [Agr95]. Segundo Lashley, uma frase falada se traduz numa seqüência neural primitiva, no caso um esquema. Contudo, a primeira definição aceita de plano vem do livro “*Plans and the Structure of Behavior*” de George Miller, Eugene Galanter e Karl Pribram em 1961:

“Um plano é qualquer processo hierárquico de um organismo tal que possa controlar a ordem na qual uma seqüência de operações possa ser executada” [GMP60], página 16.

Outro trabalho mais conhecido no contexto da IA refere-se aos estudos desenvolvidos por Allen Newell e Hebert Simon sobre a modelagem computacional na solução de problemas por meio de buscas. Os conceitos de planejamento tinham a conotação de dirigir uma seqüência de operações, de modo que pudessem desencadear uma pesquisa num espaço de estados. Eles suportavam a idéia de que a cognição humana poderia ser estruturada de forma hierárquica. Há uma decomposição hierárquica sobre a seqüência geradora de ações, de modo que disparem suas buscas locais. Em resumo, um plano contém um repertório de padrões habituais sobre uma seqüência de ações conhecidas previamente a serem executadas em situações futuras. O exemplo clássico sobre planejamento é o programa STRIPS, desenvolvido por Nilsson e Fikes [FN71].

Formalmente, um plano pode ser gerado a partir de uma *representação* de três pré-requisitos: do *domínio*, dos *objetivos* e dos *operadores*. A representação de um domínio visa uniformizar o conhecimento sobre o conjunto de seus objetos e suas propriedades invariantes. Essas propriedades não mudam, assim como as ações planejadas e executadas pelo sistema.

A uniformização tem se mostrado útil nas implementações e interações com usuários [Wil83]. A representação do objetivo deve ser congruente ao domínio modelado. O objetivo é um estado particular construído a partir dos objetos identificados no domínio. Os operadores descritos para um domínio representam as ações que o sistema pode realizar sobre os objetos. Nas ações possíveis, alguns pontos são avaliados: os objetos participantes, quais tentam atingir a meta, seus efeitos e seus pré-requisitos. Um operador possui uma descrição de como as ações mudam o estado em um domínio. Exemplo: PUTON(X,Y), colocar o bloco “X” sobre o bloco “Y”.

Após o efeito de uma ação, um planejador defronta-se com o problema do “frame”. Esse problema descreve a manutenção das consistências dos estados e uma análise da evolução do sistema. O número de relacionamentos internos começa a crescer a cada ação. Há uma estrutura de dados que deve ser mantida na avaliação desses estados.

Como se observa pelas definições acima, o problema de planejamento se reduz em encontrar um caminho de um estado inicial a um final desejado. Há uma busca por meio dos estados possíveis e pela aplicação sucessiva e sistemática dos operadores sobre os objetos do domínio. Para que se estabeleça essa busca com eficiência, é necessário um esquema de *controle* para evitar problemas como:

- A “visita” a estados já explorados anteriormente, isto é, a *ciclicidade*;
- A explosão combinatorial sobre estados já gerados (“frame problem”);
- A sistematização sobre a completude do processo. Ou seja, visitar todos estados que apresentem uma boa perspectiva de solução;
- A solução em um tempo factível.

Esses são alguns itens que a IA tem assumido no contexto de controle para planejamento convencional ou de “planejamento como um programa” [AC91]. Quanto melhor o planejamento, menor é a necessidade de controle [Wil83]. Dentro dessa sistematização, há necessidade de estruturas de dados que armazenem de forma esquemática os estados permitidos pelo sistema. Normalmente, essas estruturas apresentam-se sob a forma de *árvores* ou *grafos*. Os nós dessas estruturas representam os possíveis estados do sistema, que o plano deve avaliar ou “*visitar*”. Para que a visita aos estados do sistema ocorra, é necessário *técnicas de buscas* (ou visitas) sistemáticas sobre essa representação do problema. As descrições dessas técnicas de buscas são encontradas em [Win87, RK94, Bit96].

2.5.2 Um Exemplo de Plano

Nesta seção, delinea-se um problema da movimentação de um robô sob a visão do paradigma do planejamento clássico. Propositamente, o problema aqui discutido é resolvido nesta tese por um outro enfoque (ver capítulo 6). O problema em questão é mostrado na figura 2.1, e refere-se à movimentação de um robô móvel entre alguns blocos ou quadras.

Os arcos direcionados delineam os trechos que o robô pode caminhar. Esses arcos definem as direções permitidas na navegação do robô. O objetivo é delinear um plano que o permita partir do ponto *A* e chegar a um dos destinos dados pelos pontos *I*, *J* ou *K*. Tratando-se de arcos unidirecionais¹, as viagens de retorno não são permitidas. Ou seja, partindo-se do ponto *K*, não se alcança o ponto inicial *A*. Coincidentemente, todo desenvolvimento feito numa direção é válido como um caminho de retorno.

Por simplicidade, omite-se o conjunto das ações que o robô pode desempenhar, como “*andar para frente*”, “*virar à esquerda ou à direita*”, “*parar*”. Essas ações são omitidas em face da boa descrição lógica que esse caso apresenta. Itens que em um problema real devem ser levados em conta. Há uma analogia direta entre as ações desse problema, com as de “*mover*”, “*empilhar*”, “*desempilhar*”, “*pegar*” e “*largar*”, imprescindíveis no caso do problema do mundo dos blocos [RK94]. As ações que mudam o estado do sistema, correspondem a uma execução da mesma. A simplificação realizada só pode ser feita devido a facilidade de descrição desse domínio. A RC adotado nesse problema é o da lógica de 1a. ordem, isto é, o uso de

¹Esses arcos objetivam a simplificação na sua completude.

uma semântica de equivalência, que pode ser transformada em formas clausais com o objetivo de aplicar métodos refutativos sobre seus espaços de estados.

Para cada arco $A \rightarrow B$ é definido uma fórmula do tipo:

$$\text{trecho}(A, B).$$

Desse modo, uma especificação completa do ambiente é dada por todos os pares $X \rightarrow Y$, de modo que correspondam a fórmula $\text{trecho}(X, Y)$. Obviamente que uma seqüência de trechos estabelece um caminho. Sendo assim, um caminho é recursivamente definido como:

$$\begin{aligned} \forall_{X,Y}(\text{trecho}(X, Y) &\rightarrow \text{caminho}(X, Y)) \\ \forall_{X,Y,Z}(\text{caminho}(X, Z) \wedge \text{trecho}(Z, Y) &\rightarrow \text{caminho}(X, Y)) \end{aligned}$$

Essa descrição lógica do problema omite detalhes importantes num caso real, como posições em coordenadas de cada ponto X , orientação de sentido, distâncias de segurança dos blocos 1, 2 e 3, etc. O sentido de um movimento é o mesmo da direção, isto é, $\text{trecho}(A, B)$ implica num único caminho de A para B e não o contrário.

O exemplo acima se apresenta num domínio bem definido, com objetivos restritos de movimentação e critérios de ações embutidas no item anterior. Esse fato deve-se as uniformizações necessárias a um plano seguro. Definir um plano é validar de modo consistente os estados que se pode atingir a partir das descrições de domínios e objetivos. Esse encaminhamento ocorre por ações a serem disparadas sobre uma seqüência de estados. A abordagem aqui adotada é a lógica de 1a. ordem [CL73,

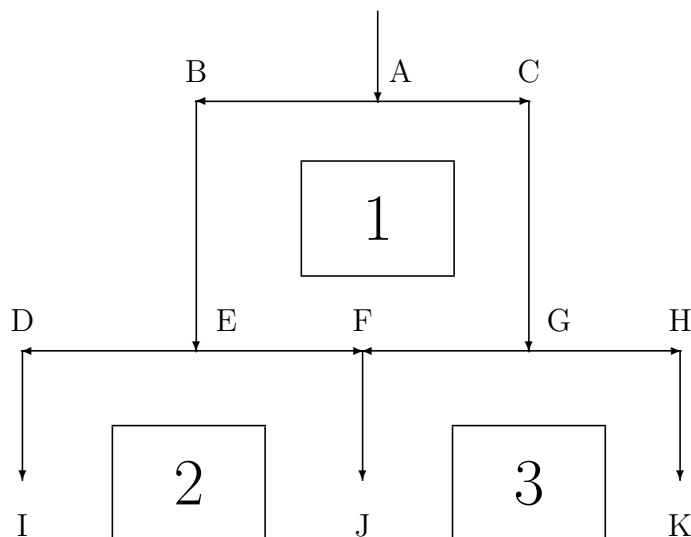


Figura 2.1: Planejamento de Locomoção

Bit96]. As “*fórmulas-bem-formadas*” (fbf) são transformadas em cláusulas de 1a. ordem e manipuladas por um provador de teorema, a partir de uma árvore de buscas de solução [RK94]. As fbf’s 2.5.1 e 2.5.2, são transcritas em uma forma clausal, que é uma notação disjuntiva normalizada. Os passos da transformação de uma fbf de 1a. ordem em uma notação clausal, podem ser encontradas em detalhes em [Bit96, RK94, CL73]. As cláusulas que definem esse problema são as seguintes:

```
/* definição de caminho */
1.  $\sim \text{trecho}(X, Y) \vee \text{caminho}(X, Y);$ 
2.  $\sim \text{caminho}(X, Z) \vee \sim \text{trecho}(Z, Y) \vee \text{caminho}(X, Y);$ 
/* definição do ambiente */
3.  $\text{trecho}(A, B);$ 
4.  $\text{trecho}(B, C);$ 
.....
10.  $\text{trecho}(F, J);$ 
11.  $\text{trecho}(H, K);$ 
```

O método de busca é a *resolução* [GC86], onde ocorre uma negação inicial do objetivo desejado, seguindo-se pela anexação à base clausal, para em seguida buscar a derivação de uma cláusula vazia. A derivação de cláusula vazia indica que a cláusula inicial era verdade, portanto podendo ser provado um caminho como solução. A *resolução* é um método de prova *refutativo*. Nega-se, inicialmente, o que é para ser demonstrado, para em seguida tentar obter uma cláusula vazia.

Para um caso propositalmente escolhido, do nó A a J , tem-se os seguintes passos das transformações clausais:

1. Nega-se inicialmente o que se deseja demonstrar, portanto: $\sim \text{caminho}(A, J);$
2. Aplica-se a cláusula 2 com as seguintes substituições: $\sim \text{caminho}(A, z_1) \vee \sim \text{trecho}(z_1, J) \vee \text{caminho}(A, J)$. Se a cláusula 1 fosse aplicada, obter-se-ia uma ramificação limitada, que pode não ser o nó desejado;
3. Cancela-se o termo $\text{caminho}(A, J)$ pelo operador “and” (\wedge), restando a cláusula resultante: $\sim \text{caminho}(A, z_1) \vee \sim \text{trecho}(z_1, J);$
4. Repete-se os passos 2 e 3 acima $n-1$ vezes, onde n representa a profundidade da árvore exploratória. Como resultado, é encontrada uma cláusula do tipo: $\sim \text{caminho}(A, z_1) \vee \sim \text{trecho}(z_1, z_2) \vee \sim \dots \vee \sim \text{trecho}(z_{n-1}, J);$
5. Aplica-se a cláusula 1 com a seguinte substituição:
 $\sim \text{trecho}(A, z_1) \vee \sim \text{trecho}(z_1, z_2) \vee \sim \dots \vee \sim \text{trecho}(z_{n-1}, J);$
6. Substitui-se cada termo z_1, z_2, \dots, z_{n-1} pelos nós B, C, E e F . A rigor, essa substituição também ocorre após o passo 3;

7. Aplica-se um subconjunto das cláusulas 3 a 11, com objetivo de obter a cláusula vazia. Por exemplo: $\sim \text{trecho}(E, F) \wedge \text{trecho}(E, F) = \square$, onde \square é a cláusula vazia.

Na seqüência, algumas observações finais são levantadas, com base no exemplo e sua relação com o planejamento clássico:

- Alguns passos foram omitidos: a rigor, há uma busca recursiva e exaustiva sobre o procedimento descrito acima. Para cada termo clausal, há uma substituição de variáveis por constantes, que visam ser eliminadas a cada passo. A cada estado gerado, todos os possíveis novos estados deveriam ser visitados;
- As descrições do ambiente são rigorosas: o ambiente descrito pode ser bem definido, segundo um modelamento, a priori de trechos direcionados. Na prática, nem sempre esse modelo rigoroso pode ser construído;
- As descrições são estáticas: relacionado ao item anterior, trechos e quadras permanecem fixas em um mesmo lugar, isto é, são válidas de modo contextual;
- Os métodos de prova crescem exponencialmente [Win87]: como planejador, utilizou-se um esquema de prova lógica. Essa escolha deve-se ao fato do problema ter “casado” com uma boa representação lógica, descrevendo-o com exatidão. Contudo, isso não o livrou dos problemas que a lógica apresenta no contexto da IA. Inicialmente, fica-se sujeito a uma *explosão combinatorial* nas árvores de buscas. Longas seqüências podem ser necessárias até que se encontre uma solução. Em segundo, tem-se o *problema da parada*, em que não há garantias que a demonstração do teorema conduza a pelo menos uma solução verdadeira;
- A descrição de um único estado é complexa [RK94]. Apesar de um simples movimento do robô mudar poucos itens no seu processo de raciocínio, uma simples alteração do ambiente faz com que as representações contextuais se alterem de forma radical. Novas regras podem invalidar ou conflitar com as já existentes, tornando necessário estabelecer um novo plano de raciocínio.

Em domínios complicados, vislumbra-se a possibilidade de decompor o problema em soluções parciais combinadas. Contudo, isso pode gerar um número elevado de combinações de soluções, tornando a questão temporal um fator limitador. Nesse caso, retoma-se o problema da controlabilidade do número de estados que um plano possa gerar. Evitar cálculos em estados já visitados tem sido uma das principais tendências na área de planejamento. Isso é, reusar um conhecimento que já se dispunha. Novamente, encontra-se diante de um compromisso com a previsibilidade do mundo, uma vez que o reuso ocorre somente na possibilidade de um estado se repetir no futuro. Como o mundo real é imprevisível [RK94], o reuso está comprometido com o armazenamento de estados já visitados, ou seja, há necessidade de sistemas com memórias a *longo prazo*.

2.5.3 Planejamento Reativo

Segundo características de planejamento tradicional, observa-se a ausência na consideração sobre as interações *momento a momento* do sistema com o ambiente. Fator que geraria um novo plano a cada variação contextual, desse modo, aplicações em tempo real tornam esse tipo de planejamento inviável. Surge uma nova proposta para contornar essas limitações, pelo *planejamento reativo*.

O termo *reativo* é caracterizado pelo estreito relacionamento estímulo-resposta imediato com o mundo, no qual o organismo reage em um baixo nível e de forma não cognitiva [Ark91]. O *planejamento reativo* (PR) visa atuar de forma flexível sob um repertório de planos disponíveis, de tal modo que as escolhas sejam realizadas em função das circunstâncias do momento [AC91].

Como aspecto negativo do PR, há o fato de não contabilizar a racionalidade global das ações desempenhadas por um sistema, um organismo ou um robô. Ou seja, a funcionalidade resultante nem sempre conduz há uma solução otimizada. Portanto, visões antagônicas surgem em torno dos planejamentos clássico e reativo, pois um privilegia o controle segundo uma avaliação global do sistema, enquanto que o segundo visa um comportamento imediato sob uma heurística local. Em resumo, as duas visões apresentam seus pontos fracos [Agr95]. Por exemplo, se a execução de um plano tiver uma curta duração de tempo e/ou exigir poucos recursos computacionais (memórias, tempo de CPU, precisão sensorial, etc), então um mecanismo de seleção ações tipicamente reativo é satisfatório. Do contrário, o reativo torna-se inviável uma vez que há necessidade de uma avaliação global de objetivo, memorizações de estados, “*backtracking*”, etc, requisitos esses incompatíveis ao PR.

Considerando que o termo “*planejamento reativo*” é ambivalente por si só, há uma contradição imediata. Essa indefinição deve ser encaminhada a uma teoria de ações, que visa explicar como uma nova ação pode assumir condições a curto e a longo prazos. Um problema que foge do escopo do tratado feito sobre PR.

Não há objetivos em contemplar ou reconciliar essas duas visões, mas sim apresentar ações flexíveis ao instante considerado. De acordo com P.E. Agre, essa tentativa de combinar a reatividade a um planejamento clássico seria uma proposta ilógica e sem perspectivas².

Os sistemas que apresentam o PR são considerados como sistemas de “*arquiteturas híbridas*” [Ark91, Agr95]. Contudo, pouco se sabe de implementações que tenham atingido um bom grau de satisfação quanto a seus resultados [Agr95]. Esse tipo de planejamento pressupõe visões contraditórias quanto à ação disparada. Detecta-se a ausência de uma estrutura conceitual consistente, que combine essas duas abordagens de planejamento. Desse modo, aplicações práticas podem estar comprometidas pela ausência de uma estrutura que seja factível, e o mais importante: ser completa (ou quase), sob essas duas visões. Um plano a ser construído sobre essa visão leva o estigma de ser improvisado momento a momento, objetivando um alto grau de flexibilidade sobre cada ação [Agr95]. Atualmente, as pesquisas sobre planeja-

²Mensagens trocadas por email.

mento estão concentradas sobre o “processo de construção” de planos; a execução dos planos tornou-se um problema menor do que a sua geração automática [Agr95]. Há uma questão biológica apontada pelos pesquisadores: a adaptabilidade de seres vivos primários, tipo uma ameba, exibem comportamentos (seqüência de ações) mais interessantes que sistemas baseados em paradigmas da IA. Surge o questionamento: até onde vale a pena construir e executar planos em sistemas do tipo autônomos? Parte dessa pergunta é respondida nesta tese.

Capítulo 3

Apresentação à Inteligência Artificial Distribuída

3.1 Introdução

Este capítulo visa definir conceitos pertinentes a agentes e motivações em torno da Inteligência Artificial Distribuída (IAD). A primeira seção fornece uma panorâmica e razões que levaram ao surgimento da IAD. Em seguida, duas tendências atuais de abordagens, que dividem a IAD são encaminhadas: a Solução Distribuída de Problemas (SDP) e a de Sistemas Multiagentes (SMA). Na seções que se seguem, é apresentada uma arquitetura genérica de agente, especificando-se dois tipos importantes: os agentes cognitivos (AgC's) e os agentes reativos (AgR's). Finalmente, utilizando-se de figuras gráficas, são ilustrados vários elementos e conceitos na interação entre agentes. A visão aqui discutida sobre a IAD é dirigida ao corrente trabalho.

3.2 Discussão Geral

3.2.1 Motivações Preliminares

A IAD surgiu na década de 80 como uma sub-área da IA clássica. Pesquisadores como Nils Nilsson acreditam que a IAD tenha um papel fundamental no entendimento da IA. Tal afirmação se baseia em [Cd96]:

- Os sistemas estão muito complicados e com muito conhecimento embutido. Permitindo-se uma “quebra” desses sistemas em partes menores, como entidades cooperativas, haveria a um aumento de eficiência;
 - Esta decomposição promove uma estrutura base, que conduz a melhores testes sobre conhecimento, baseados em processos de raciocínio, ações e planejamento;
 - Os métodos usados em sistemas inteligentes para raciocinar sobre um sistema de ação, também podem ser usados em outros processos dinâmicos não-inteligentes;
-

- Há um melhor entendimento nos processos de comunicação entre as entidades cooperativas e até mesmo na comunicação homem-máquina, tal como a linguagem natural.

Considerando as tecnologias já sedimentadas na IA clássica, vislumbrou-se ainda que a IAD exerceria um papel integrador neste contexto. A questão é como reunir de forma harmônica todo conhecimento sobre IA já disponível. Como a IA já dispõe de técnicas robustas dirigidas a sistemas inteligentes, mas específicas, a questão é como ampliar esse potencial para diversos contextos. Portanto, combinar coerentemente essas partes passou a ser um nicho de pesquisas em IAD [Dur91].

Alguns sistemas são naturalmente complexos e distribuídos em seus ambientes. Eles apresentam um “gargalo” (uma limitação) espacial e/ou temporal, segundo uma ação de resposta a um estímulo. Exemplificando: numa rede de abastecimento e distribuição elétrica, a ação disparada, devido à queda de um dos nós, torna necessária uma visão global de toda rede, bem como local. Há uma necessidade de se considerar um balanceamento global da carga, segundo recursos disponíveis nas proximidades, sem detrimento dos outros pontos de distribuição de carga. De modo cooperante, também se necessita de uma rede de comunicação eficaz, levando-se em conta a inerente distribuição geográfica que é a rede de energia elétrica. Enfim, a IAD dirige-se a questões que envolvem o aspecto da distribuição, da coordenação do conhecimento e ações em ambientes que envolvam múltiplas entidades¹ [Cd95].

Nesse comprometimento entre muitas entidades, surge a questão dos recursos em um sistema real, onde são inerentemente limitados. Esses recursos, normalmente, são compartilhados por mais de um usuário e/ou agente² [CJ93]. Em uma interessante discussão, R.M. Turner [CJ93] conduz o problema de recursos compartilhados em torno das questões ecológicas. A base é a existência de uma rede de implicações nos ecossistemas. Todo recurso tem uma capacidade nominal, uma quantidade máxima que pode suportar, considerando uma população de provedores versus uma de usuários. Há um compromisso de busca a um equilíbrio, evitando assim, uma degradação sistêmica.

Em uma de suas definições, a IAD é caracterizada pela interação de diversos agentes, visando atingir um objetivo global de maneira cooperativa e não conflitante [Hog91, Gas89]. A proposta é que agentes possuam um certo tipo de conhecimento e por uma interação conjunta estejam empenhados em resolver uma tarefa comum ou não. Nesse nível, um sistema forma uma *sociedade de agentes* [DB92]. Les Gasser [Gas91] faz referência a uma “comunidade” inteligente formada por uma *coleção de interações, coordenadas por processos baseados em conhecimento*.

Esse encaminhamento de Les Gasser [Gas91] é motivado pelo fato do conhecimento de uma atividade interessante em alguns domínios ser “*inerentemente distribuído*

¹Oportunamente, essas entidades são definidas como *agentes*. Um conjunto de agentes pode ser visualizado como uma *sociedade*. Os agentes têm seus objetivos particulares, bem como a sociedade por inteira tem a sua meta.

²Esse termo é detalhado em outras seções.

espacialmente”. Essa é uma razão em se desejar uma inteligência distribuída ou copiá-la por um sistema de *multiagentes*³.

3.2.2 Quanto à Distribuição

Dentro desse aspecto de *distributividade* dos sistemas, L. Gasser [Gas91]⁴ advoga que isso leva a critérios motivadores de projetos no que tange a: ... “*adaptabilidade, redução de custos, facilidades de desenvolvimento e manutenção, aumento de velocidade, necessidade de autonomia e especialização*”.

Por via de regra, todo sistema é intrinsecamente distribuído (“*All real systems are distributed*”) [Gas89]. Por conseguinte, a modelagem de suas soluções segue essa tendência de distributividade e conectividade entre as partes (subsistemas). Nessa linha, a IAD utiliza-se do conceito de agentes na negociação e coordenação das partes de um sistema. A coordenação envolve questões de coerência e coesividade entre as partes, com objetivo de se obter um comportamento desejado.

Nesse ínterim, os problemas reais abordados tornaram-se mais complexos, bem como os métodos de soluções. Com isso, surge a necessidade de tecnologias que abordem os problemas com ênfase na cooperatividade e distributividade entre as possíveis soluções. Isto é, como integrar vários componentes de naturezas distintas, de modo cooperante e não conflitante?

Evidencia-se a necessidade de acoplar harmônicamente elementos (módulos, subsistemas, etc) de forma a cooperarem para um objetivo global. Esses componentes tornaram-se peças *ativas* nas soluções globais. Com isso, comumente denominou-se por *agente* um módulo com funções específicas, que possui um conhecimento interno e externo, com uma possibilidade de diálogos interativos com outros agentes de uma vizinhança. Essa definição é melhor explorada ao longo do texto.

Num segundo momento, surge a necessidade de se gerar um padrão conceitual em torno dos termos da IAD. Trata de um segmento que explora a cooperatividade de entidades (agentes), inicialmente apoiada numa metáfora sobre a *interação social das pessoas*. A investigação de conceitos para um único agente e suas ações envereda-se por um caminho de desconexão e descomprometimento sistêmico. Outras metáforas foram utilizadas, como competitividade comercial, pesquisa na comunidade científica, Biologia, Sociologia, etc [Zie92]. O enfoque metafórico desse capítulo é o social. Contudo, questões biológicas são consideradas, tendo em vista esta tese.

Por razões históricas, dividiu-se a IAD sob dois enfoques: Solução Distribuída de Problemas (SDP) e Sistemas Multiagentes (SMA). Cada uma dessas visões fornece um repertório específico para seus termos, levanta diferentes questões na solução de um problema e distintas maneiras de abordá-los. Contudo, essa divisão não é consensual, mas é um início para se delinear a IAD.

³Esse termo é traduzido e aqui adotado do original em inglês “*multi-agent*”.

⁴Essa referência é clássica na área de IAD; um ponto de partida para conceitos e teorias.

3.3 Solução Distribuída de Problemas (SDP)

A SDP é caracterizada pela ausência de um esquema decisório centralizado ou uma autoridade de controle. Há uma rede que objetiva completar todas suas tarefas com ações cooperativas entre os seus nós [Les91]. Esses nós são agentes pré-programados, cujo objetivo é uma cooperação robusta e eficiente do problema distribuído nessa rede. O principal foco da SDP é o *problema*, cujo enfoque é derivativo da IA simbólica. Desse modo, o problema se resume em decomposição e alocação de tarefas, segundo uma estrutura de redes [Bit96].

Resumidamente, o processo da SDP pode ser visto em três etapas [Cd95]:

1. O problema é decomposto em subproblemas: o processo da decomposição pode envolver um particionamento hierárquico ou mesmo em nível;
2. A solução dos subproblemas: essa solução é realizada por agentes situados que cooperam e se comunicam se necessário for;
3. A integração: os resultados dos subproblemas são integrados com objetivo de se obter a solução completa ou total do sistema.

Observa-se que a SDP deve dispor de múltiplas fontes de especialistas, múltiplas habilidades e múltiplas perspectivas [DR94]. As perspectivas correspondem a problemas inerentemente distribuídos, como controle de tráfego aéreo, redes de comunicação, monitoramento distribuído de veículos, redes de sensores etc; e como as diversas soluções (as visões externas e internas) interessam mutuamente aos participantes dos problemas. O exemplo típico de uma abordagem via SDP pode ser ilustrado por múltiplos sensores, distribuídos no corpo de um avião. Cada sensor possui uma visão local e restrita sobre o controle do avião, embora uma fusão dessas múltiplas perspectivas sensoriais permita um controle eficiente com uma visão global do avião.

A SDP deve apresentar uma coerência entre as múltiplas entidades envolvidas. Uma vez estabelecido um comportamento resultante desejado, essas entidades devem apresentar sinais de concordâncias, compartilhamento de dados comuns, comunicação, etc. Apesar da existência de algum tipo de conexão entre as múltiplas entidades, a SDP é caracterizada por um fraco acoplamento entre essas entidades. O acoplamento é restrito, fraco, de baixa interação, permitindo as entidades as condições necessárias e suficientes ao compartilhamento de recursos comuns.

3.3.1 Um Exemplo

Um exemplo típico de SDP é reportado por Conry e outros [Les91]. O trabalho descrito em [Les91] aborda um paradigma de negociação em multiestágios (NM) para SDP com satisfação de restrições. A negociação em multiestágios (NM) é um protocolo que generaliza o das *redes contratuais* [Smi83]. As redes contratuais ocorrem quando um único agente efetua a decomposição do problema e, em seguida, negocia

com os outros agentes para determinar quais são as subtarefas destinadas a cada agente. No protocolo de negociação em multiestágios há uma contínua iteração de mudanças de definições e opiniões por parte do nó (o agente) sobre o seu impacto de escolha de tarefa local, de modo que seu desempenho contribua para solução global do problema. Há uma semelhança com o paradigma de *funcionalidade precisa/cooperativa* [Cor81, CL95, Car95] (“FAC – *functionally accurate/cooperative*”) onde os nós mudam iterativamente em suas explorações e tentativas. Contudo, em NM são considerados os melhores resultados parciais das subtarefas locais. Essa estratégia de cooperação permite que soluções sejam incrementalmente construídas, com o propósito de convergir para um conjunto completo de soluções, que sejam globalmente consistentes. Em resumo, há uma busca distribuída de soluções locais, nas quais se verifica, constantemente, a incompatibilidade entre elas na solução global do problema. A combinação dos recursos locais e da coordenação requerida pelas ações entre os nós resultam num complexo conjunto da solução global, independente das restrições locais.

O protocolo de NM é apresentado em [Les91] sob a forma de um algoritmo de três fases. A primeira fase ou de busca *assíncrona* é iniciada identificando os conjuntos de agentes com ações conjuntas que satisfaçam os muitos objetivos do sistema. A segunda fase ou de busca *coordenada* visa encontrar uma rede de solução que respeite os conflitos não locais de cada agente, isto é, considera as vizinhanças dos agentes. A terceira fase é verificar se o problema se encontra *fechado* (limitado ou restrito) em suas soluções globais, selecionando um plano adequado a pelo menos uma execução do objetivo global. Novamente, nessa fase, os agentes entram em negociação na tentativa de estabelecer subplanos, que respeitem seus recursos visando uma solução global.

O estudo de [Les91] é feito com um exemplo das redes de computadores com diversos nós processadores espalhados em uma área geográfica. O problema é interessante pelo simples fato de que cada nó é um agente semi-autônomo resolvidor de problema, com um conjunto finito de recursos locais, e uma limitada capacidade de comunicação entre os outros nós. Define-se o agente como responsável pelo controle de fluxo em uma sub-região geográfica desta rede de computadores. Desse modo, cada agente é constituído por um nó de rede ou por um conjunto de nós. Cada agente tem como requisito certos recursos, que são os canais de comunicação. Para que um objetivo global seja atingido, deve haver uma entrega de mensagem entre dois pontos da rede. Portanto, há vários pedidos que são requisitados como serviços da sub-rede de comunicação. O detalhe é cada agente reconhecer seus recursos nas vizinhanças. Não há planos globais e explicitados para cada agente quanto às rotas que uma mensagem assume na rede. Cada agente é definido em termos de um conhecimento local, onde se conhece os objetivos e subobjetivos, e os recursos disponíveis. Considerando a queda de um recurso, uma indisponibilidade, rotas alternativas devem ser estabelecidas nesse sistema distribuído. O grafo de restrições é alterado para que o objetivo seja satisfeito. Enfim, é um problema que envolve múltiplas dependências e que pode envolver vários passos de negociação para detecção e resolução de um problema.

De acordo com [Bit96], são quatro as principais classes de problemas abordados

pela SDP: Interpretação Distribuída (redes de sensores, diagnósticos de falhas em redes em geral, etc); Planejamento e Controle Distribuído (controle distribuído de processos em manufatura, alocação em redes, etc); Sistemas Especialistas Cooperantes (veículos autônomos, negociação entre especialistas, etc); e Cooperação Humana via Computador (“groupware” - coordenação de projetos, editores distribuídos, etc).

3.3.2 Resumindo

Do exemplo acima, observa-se que a abordagem pela SDP possui uma rede interligada de nós, fracamente acoplada, que cooperam coerentemente na solução de um problema complexo. Os principais objetivos da SDP são resumidos em [Bit96]:

- Aumento da eficiência devido ao paralelismo desencadeado pelos nós;
- Aumento do número de tarefas realizáveis pelo compartilhamento de recursos (informações, conhecimento, recursos físicos). Há uma reusabilidade dos recursos comuns [Tur91];
- Aumento da confiabilidade (tarefas duplicadas são executadas por métodos diferentes);
- Diminuição da interferência entre tarefas (nós), evitando-se interações inúteis. Apenas as mensagens pertinentes à conexão são consideradas relevantes no processamento do agente.

Finalmente, todo esforço da SDP tem como ênfase o *problema* e como múltiplas entidades inteligentes (computadores programados) podem resolvê-lo conjuntamente. Tudo isso, sob uma boa perspectiva de eficiência, segurança, coerência e robustez.

3.4 Sistemas Multiagentes (SMA)

Na proposição básica da IAD sobre os agentes que resultam numa ação cooperativa interessante à sociedade, a abordagem por SMA tem como foco de estudo o *agente* e sua perspectiva social. O enfoque é sobre a sua capacidade de relacionar-se com os demais agentes, visando uma ação comum. Em SMA, os agentes apresentam uma certa autonomia. Eles são potencialmente pré-existent e tipicamente heterogêneos [Cd95]. Outro aspecto a ser levantado em SMA, é a questão de suas arquiteturas de controle. Em [Bit96], há um resumo de quatro propostas. São elas: Sistemas “Blackboard”, SMA Federados, SMA Democráticos e SMA Abertos. Este item reflete como os agentes interagem em SMA. Ao contrário da SDP, aqui não há uma topologia rígida sobre a qual os agentes se comunicam.

A abordagem por SMA visa à coordenação de um comportamento inteligente entre os vários agentes da comunidade. A questão é como os agentes coordenam seus conhecimentos, seus objetivos, suas habilidades e planos, para em conjunto, tomarem

uma ação para resolver problemas. Nesse tipo de ambiente, os agentes podem estar trabalhando sobre um objetivo global ou visando seus objetivos individuais. Igualmente aos nós na SDP, os agentes podem compartilhar recursos, como conhecimentos e resultados parciais. Diferentemente à SDP, os agentes na abordagem SMA devem também raciocinar sobre o seu processo de coordenação [Jen96]. A coordenação em SMA é um aspecto crucial. Sem ela, qualquer benefício de interação inexistente e o comportamento do grupo de agentes pode mesmo tornar-se caótico [Cd95].

3.5 Relação entre SMA e SDP

De modo oportuno, questiona-se qual a relação existente entre essas duas abordagens básicas na IAD. A relação entre SMA e SDP não é muito clara e, eventualmente, pode mudar ao longo do tempo [Bit96]. Em [DR94], essa relação é apresentada sobre três visões não mutuamente exclusivas, estabelecidas uma sobre a outra com alguma variação:

1. Visão 1: A **SDP** é um subconjunto de **SMA**;
2. Visão 2: Os **SMA** fornecem uma base (um substrato) para **SDP**;
3. Visão 3: **SDP** e **SMA** são linhas de pesquisas complementares.

A discussão dessas visões são encontradas em [DR94] e [Bit96]. Essa é encaminhada em termos de suposições sobre a SDP e os SMA, como a benevolência⁵ entre agentes, objetivos comuns, projetista centralizado, decisões individuais versus um comportamento global, propriedades internas de agentes, planos locais (parciais) e globais, convergência de objetivos, etc.

Finalmente, E.H. Durfee e J.S. Rosenschein [DR94] concluem que as áreas SDP e SMA estão se ajustando mutuamente num encadeamento conduzido da individualidade do agente ao sistema global. Sob o risco de serem simplistas, Durfee e Rosenschein [DR94] afirmam que a IAD envolve *agentes* que agem em um *ambiente*, formando juntos um *sistema*. Então, uma análise sob a visão da IAD implica em estudar as propriedades dos *agentes*, dos *ambientes* e dos *sistemas*. As propriedades implicam em questões como:

- Agente: Quais são suas habilidades? É racional⁶? Reconhece os demais agentes? Como são suas preferências quanto a recursos comuns?
- Ambiente: É estático? Fechado? Benigno? As ações resultantes dos agentes são predizíveis? Há variações temporais? É temporariamente limitado?

⁵Um agente benevolente é aquele que deseja ajudar um outro agente sempre que possível.

⁶Um agente é dito ser racional quando age no sentido de maximizar seus benefícios e minimizar suas perdas.

- Sistema: O que se pode dizer do sistema agente/ambiente? As propriedades internas, como acesso a recursos com igualdade, honestidade entre os agentes, etc, são asseguradas? As propriedades externas, como respostas temporais para as entradas do sistema, são fixas?

Propriedades			
	Agentes	Ambiente	Sistema
SMA	variável	fixa	fixa (interna)
SDP	fixa	variável	fixa (externa)
?	fixa	fixa	variável

Tabela 3.1: Temas de Pesquisas (SDP,SMA) e Propriedades

Segundo os tipos de propriedades desses três itens, a tabela 3.1 sumariza a sua interseção com as abordagens SDP e SMA. Novas considerações sobre SDP e SMA podem ser feitas a partir da tabela 3.1.

SDP: Considera as propriedades externas, como a robustez e a eficiência, sob variações ambientais, a partir de agentes com propriedades bem estabelecidas. A SDP visa encontrar uma coleção particular de agentes com algum nível de desempenho coletivo, sendo as propriedades do ambiente dinâmicas e incontroláveis;

SMA: Considera certas propriedades internas de um sistema de agentes cujas propriedades individuais podem variar. Os SMA têm focalizado as questões de como os agentes com preferências individuais interagem entre si, em ambientes particulares, de modo que a concordância em agir, conduza a propriedades globais desejadas.

A categorização acima é apenas um ponto de partida sobre abordagens na IAD. Contudo, Durfee e Rosenschein [DR94] deixam em aberto a categorização das propriedades de agentes e ambientes quando fixas e as propriedades dos sistemas variáveis. Como possíveis candidatos a essa categoria, citam-se as redes neurais e a computação evolutiva. Na prática, a maioria dos sistemas não é classificada em uma abordagem pura da SDP ou dos SMA, mas sim mescladas. Porém, ocorre a predominância de uma abordagem ou outra, evidenciada pela quantidade de parâmetros que se relacionam a um projeto de IAD.

3.6 Agentes em IAD

3.6.1 Em direção a um conceito de agente

Um dos conceitos fundamentais da IAD é o de *agente*. Outros conceitos de mesma importância na IAD são os de *arquitecturas* e *linguagens* [WJ94]. Em uma primeira instância, o agente é uma entidade que pode ser descrita por suas ações externas, racionais e intencionais, de acordo com seus objetivos e percepções externas. Essas percepções refletem um conhecimento de seu estado corrente [DB92]. Alguns fatores do agente, como as percepções, interação com outros agentes, intenções, estado de crença do seu modelo interno, etc, tornam o agente apto a realizar uma ação. Contudo, a determinação do que é uma ação gerada por uma percepção externa pode encaminhar o problema para uma visão filosófica. Esse ponto é ilustrado a seguir.

São muitas as propostas e as caracterizações sobre agentes. A literatura nesta área é rica e continua a crescer buscando novas definições teóricas e práticas, novos modelos arquiteturais, etc. A questão reflete sobre que *visão* se deseja definir um agente. Qual o enfoque a ser considerado: o computacional, o biológico, o social, o psicológico etc? Para definir, são levantados pontos como: é sobre o seu conhecimento estático, embutido ou sobre um conhecimento temporário e volátil (dinâmico)? É sobre sua comunicação explícita ou implícita com outros agentes? Qual o propósito do agente em um dado contexto? Qual a sua representatividade? Qual a sua complexidade interna? Isso leva a associar o agente a uma partícula elementar como um *grão*. Surge um primeiro conceito em torno do agente: a sua *granularidade*. Nesse contexto, granularidade refere-se à fração do agente em relação à comunidade multiagentes. Ou ainda, reflete um compromisso entre um grau de autonomia local e uma cognição global. Exemplificando, em um esquema neural a granularidade é equivalente ao neurônio biológico. Esse é um tipo de agente biológico bem definido em um contexto. Contudo, uma analogia com programas computacionais, tornaria essa caracterização um pouco problemática. Um agente pode ser uma rotina destinada ao reconhecimento de uma dada figura geométrica complexa? Um agente pode ser um simples comando “*exit for*”? Nas duas questões, acreditamos que a resposta seja positiva. Nesse caso, as definições do agente são malcomportadas, mal-definidas, etc. São granularidades diferentes, mas com um papel ativo no processo. Enfim, observa-se pela literatura [WJ94, Cd96] e pelos exemplos acima que a definição de agente é **contextual aos domínios**.

Buscando novas definições, M.J. Wooldridge e N.R. Jennings [WJ94] levantam o ponto de que agentes possuem um papel ativo, dando origem a ações que afetam os ambientes e não o contrário. Sob essa visão, a interação com o ambiente é enfraquecida, perdendo os oportunismos oferecidos aos agentes [Bro91c, Bro91b]. Em torno da ação, duas definições são pertinentes: *autonomia* e *racionalidade* [WJ94]. A autonomia geralmente significa que o agente opera sem uma intervenção humana explícita. A racionalidade visa maximizar o desempenho do agente em relação a alguma função de avaliação. Essas definições colocam as definições de agente em uma ampla faixa de objetos. Por exemplo, uma ação de um transistor (menor unidade lógica de chavea-

mento eletrônico) em um circuito exhibe as duas características mencionadas. Portanto, objetos como termostatos, objetos de linguagens de programação orientada á objetos, entidades neuronais, sensores-atuadores etc, são agentes simplificados. Por outro lado, em [WJ94] discute-se a conceitualização do agente exhibir uma cognição de “alto nível” típico da IA clássica, condição que invalida os exemplos acima e que leva a detectar um espectro na categorização de agentes. A existência desses diversos tipos de agentes, com diferentes granularidades, vai ao encontro da questão contextual do agente.

3.6.2 Um Modelo de Agente

Neste ponto é apresentado um modelo genérico de agente, que pode ser simplificado, ou ampliado a um agente de “alto nível”. Em algumas extensões sobre o modelo aqui reportado, o agente pode se tornar complexo e habilitado a tarefas complexas (por exemplo, o reconhecimento de padrões). O objetivo desse modelo neste momento do trabalho é ilustrar os pontos levantados anteriormente, bem como outros que seguem.

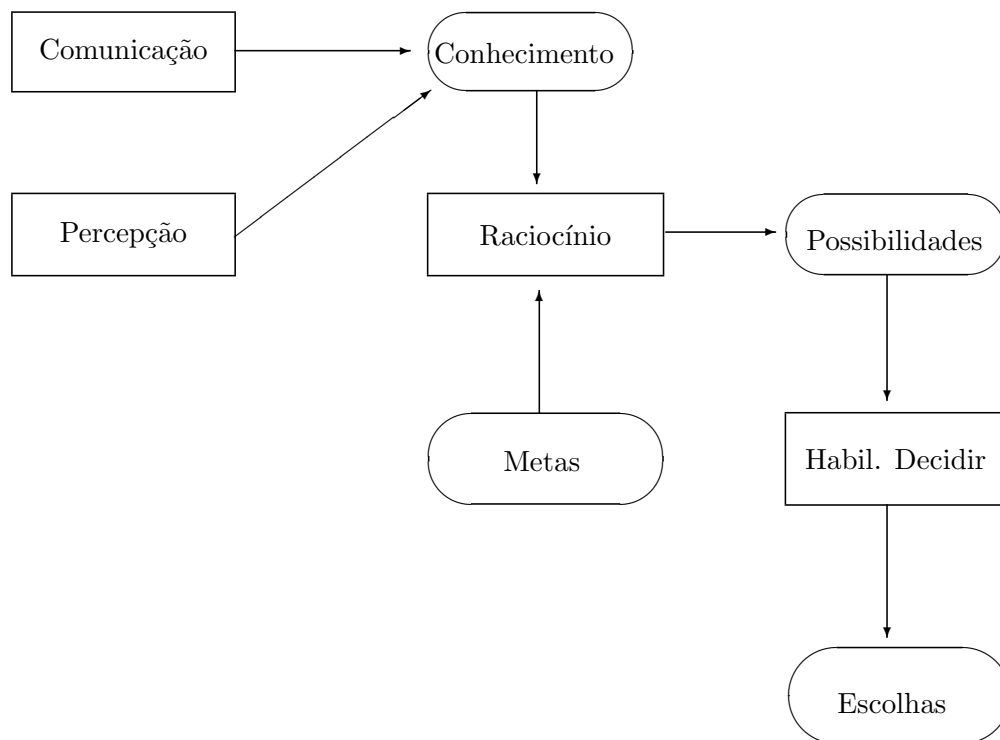


Figura 3.1: Modelo Genérico de Agente

O modelo de agente escolhido vem dos trabalhos de Yves Demazeau e outros [DM90, DM91], reapresentado na figura 3.1. O modelo da figura 3.1 é aproximadamente consensual e possibilita explorar os principais conceitos propostos pela IAD [Cd96, Cd94, Zie92, BD94]. Esse modelo é genérico no que concerne a interações com

outros agentes e a função desempenhada pelo agente. O modelo de agente genérico de Yves Demazeau é interessante pelo fato de permitir que sofisticções e reduções sejam feitas sobre essa estrutura conceitual.

O módulo do *conhecimento* possui as representações implícitas e explícitas que o agente deve ter sobre o mundo ou de seu objetivo (local e global). Esse conhecimento pode ser dado ou adquirido pela *percepção* ou pela *comunicação* com outros agentes [DM90] ou ainda com o ambiente.

O módulo de *metas* não precisa ser necessariamente explicitado. “*As metas podem ser codificadas por algoritmos, ou explicitamente definidas, ou adquiridas pelo contexto, ou obtidas pela comunicação com outros agentes*” [DM90]. Suas metas podem ainda ser deduzidas pela observação de um comportamento característico do agente.

Os módulos de *conhecimento* e *metas* estão entrelaçados pela necessidade do agente esboçar um plano ou um raciocínio de como chegar a um objetivo com o conhecimento disponível. Há um esquema que associa esses dois módulos para se inferenciar uma *possível solução*. Então, a *habilidade de raciocínio* leva à derivação de algumas das possíveis soluções [DM90].

Quando várias respostas são passíveis de ser aplicadas, necessita-se escolher a(s) mais apropriada(s). Contudo, o agente pode, e normalmente deve, possuir suas limitações internas. As limitações levam a um gerenciamento de recursos disponíveis. Desse modo, surge a necessidade de um módulo que defina a *habilidade de decidir* a solução mais rentável num dado contexto. A melhor solução é escolhida segundo o ponto de vista interno do agente. “*Devido as limitações internas do agente, a escolha pode ser errada segundo um ponto de vista externo*” [DM90]. Finalmente, no módulo *escolha* infere-se a ação do agente.

A comunicação entre agentes é um ponto marcante na IAD [BD94]. Contudo, o seu processo é complexo e ainda em aberto. Um exemplo formal sobre esse processo é apresentado em [DB92]. A comunicação entre agentes ocorre por módulos de *percepção* e de *comunicação*. As informações trocadas entre os agentes são de três tipos: *conhecimento* (I), *possíveis soluções* (II) e *escolha* (III). Em face desses tipos, o processo da comunicação entre os agentes dar-se-á em níveis do tipo *forte*, *médio* e *fraco* [DM90]. A heurística nesses níveis segue o número de tipos de informações trocadas. As interações classificam-se em:

I, II e III \Rightarrow forte: A interação é forte no sentido de apresentar uma influência na habilidade de decidir sobre a escolha do agente. Ou seja, a escolha na saída do agente é afetada.

I e II \Rightarrow média: A interação é média no sentido do agente considerar uma influência de um outro agente na sua habilidade de raciocínio.

I \Rightarrow fraca: A interação é fraca no sentido do conhecimento ser comum e exposto em igualdade para todos os agentes. Nesse caso, há um fraco acoplamento entre os agentes.

Quando as informações se originam do mundo externo, seguem através da percepção. Caso as informações venham de outros agentes, o módulo da comunicação é responsável por essa tarefa. Tratando-se de implementações computacionais, o módulo da comunicação pode se utilizar de paradigmas, como ator⁷, “blackboard”, etc [DM90]. Vale ressaltar que I, II e III refletem estruturas, descrições, mensagens abstratas nos vários níveis do agente.

A partir desse modelo genérico de agente, há duas categorias de agentes que podem ser instanciadas com simplificações ou extensões, conforme descrito no início da seção, ou sejam, os agentes *cognitivos* e *reativos*. A escolha deve-se a sua pertinência com alguns trabalhos da IAD e desta tese.

3.6.3 Agentes Cognitivos (AgC's)

Um agente cognitivo tem a habilidade de realizar um certo raciocínio sobre um modelo computacional mais sofisticado. Normalmente, esses agentes exibem métodos poderosos de uma computação interna e uma sofisticada comunicação externa com outros agentes ou com o ambiente. Essa computação interna reflete a competência do agente em realizar uma determinada tarefa.

Nos AgC's, a representação e raciocínio simbólico fazem uso de técnicas já consolidadas da IA clássica, como a dedução, o planejamento e a aprendizagem [OD94]. Em face da disponibilidade dessas diversas técnicas, considera-se os AgC's generalistas sob várias visões, contudo apresentando uma desvantagem, quando se trata de processamentos em tempo real [OD94]. A incorporação dessas sofisticadas técnicas nos AgC's, faz com que fiquem lentos, devido à complexidade de seus algoritmos, bem como falhos sob eventos imprevisíveis.

Um conjunto típico de componentes de um AgC é apresentado na figura 3.1. Vale ressaltar a existência de uma memória para o processo de raciocínio e a manutenção de um histórico de estados. Dentro do processo de raciocínio, os itens como *crença*, *intenção* e *desejo* são considerados pelos AgC's [WJ94, BD94]. Esses itens são pertinentes à cognição humana. Desta forma, os projetos de AgC's levam em conta fundamentos apresentados em modelos de organizações sociais humanas, hierarquias de grupos, relacionamentos formais, modelos psicológicos, etc. Tendo em vista que um dos objetivos da IAD é tornar o ser humano participante do sistema, os AgC's fazem o papel de elo nesse processo cooperativo. Atualmente, os AgC's ocupam a maior parte das pesquisas na IAD. Referências sobre alguns trabalhos são encontradas em [WJ94, Cd94, Cd95, Cd96, Bra94].

3.6.4 Agentes Reativos (AgR's)

Os modelos de agentes reativos apresentam uma forte inspiração biológica. Mais especificamente na etologia animal. O exemplo típico de um agente reativo é o da formiga caminhando em uma praia [Bro91d]. Evidentemente, a formiga não possui

⁷Cada objeto tem vida independente e comunica-se com os demais por mensagens primitivas.

um conhecimento sobre o relevo da praia, porém possui um eficiente mecanismo de adaptação na sua relação sensorial-motora. “...*comportamento da formiga é mais um reflexo da complexidade do ambiente do que da sua própria complexidade interna*” [Bro91d]. Cada novo relevo é desconhecido pela formiga porém ela gera uma ação motora adaptada e robusta ao contexto corrente. A motivação desses agentes é seu comportamento emergente⁸, uma vez que apresentam uma contínua interação com o ambiente, e rápidas respostas como reação.

Um agente reativo tem a habilidade de reagir sob contextos inesperados. Sua reatividade ocorre com as mudanças do ambiente ou ainda de uma comunicação com outros agentes. Para ele, não há nenhum artifício que informe sobre algum conhecimento específico do ambiente. Exceto o que ele obtém de sua entrada sensorial ou de uma classe de estímulos sensoriais.

Uma das características desses agentes é a rápida seleção da ação com base nessas percepções [OD94]. A interpretação das informações sensoriais de seu ambiente é quase que imediata seguindo regras muito simples. A associação entre as entradas perceptuais com as saídas efetoras programadas ou via “*hardwire*” são, normalmente, expressas sob a forma de regras e organizadas numa arquitetura em multicamadas [BD94, Bro91c, Bro91a]. Os conflitos gerados nos disparos simultâneos das regras, geralmente, são resolvidos por alguma ordem de prioridade.

Agentes reativos não raciocinam sobre suas intenções, isto é, não há uma manipulação em torno dos dados visando uma meta [Cd96]. As ações executadas pelos AgR's resultam do disparo de regras ou da execução de planos estereotipados. Os Sistemas Especialistas de primeira geração (base de fatos, conhecimento explícito num conjunto de regras e um motor de inferência) são os exemplos típicos de AgR's. Dentro de um esquema SMA, os AgR's possuem a habilidade de se comunicar com os outros agentes, enviando, recebendo, escolhendo e interpretando relevantes mensagens no contexto da comunidade de AgR's [Cd96].

Diante aos eventos imprevisíveis, os AgR's devem possuir características de adaptação comportamental. Os imprevistos inerentes a um ambiente real, levam há dois aspectos a serem considerados na construção dos AgR's [OD94]:

- A percepção deve ser suficientemente rápida para que a “imagem” (fotografia dos sensores ambientais) capturada seja compatível com o real contexto corrente;
- A computação e a execução da ação a ser disparada devem estar comprometidas com o estado corrente.

Observa-se que geralmente os AgR's não possuem memória ou ainda um sofisticado modelo simbólico para realizar raciocínios⁹ como os AgC's. A inexistência de memória não os permite manter um registro histórico de seus passos. Essa ausência de memória e de sofisticados processamentos, tornam os AgR's atrativos sob a visão do

⁸Ver [Ste94b].

⁹Nesse contexto, define-se esse raciocínio por *controle deliberativo*.

seu custo de projeto, tornando-os interessantes quando existentes em grandes quantidades. No trabalho de L. Steels [Ste88], há uma simulação de um grupo de agentes explorando um planeta. Cada AgR possui um simples conjunto de regras, que os habilita a coletar amostras deste planeta e retornar à nave-base da missão. Com relação a esse sistema (um conjunto de AgR's e ambiente), tem-se, como principal vantagem sobre abordagens clássicas, o paralelismo e a concorrência dessa exploração planetária [Fly89].

3.7 Comunicação entre Agentes

3.7.1 Tipos de Comunicação entre Agentes

A cooperatividade evidenciada na IAD faz com que mecanismos de comunicação sejam necessários entre agentes de uma comunidade. Os agentes precisam se comunicar, visando seus objetivos locais ou, ainda, convergindo para um objetivo global. A comunicação entre agentes esclarece as intenções individuais perante o contexto social, bem como a existência de uma coordenação das atividades. O processo da comunicação em IAD, normalmente, é realizado por um esquema de *protocolos* [Ros94].

Protocolo é um conjunto de regras que normatiza e controla a troca de informações entre os agentes. O agente deve reconhecer qual o protocolo estabelecido pela sociedade para que esteja habilitado em interagir com os demais agentes. Um protocolo torna possível a participação social por parte de um agente, permitindo que efetue restrições e contribuições à sociedade ou, ainda, enriquecendo a si próprio [Jen96]. Enfim, o uso de protocolos é um oportunismo por parte da IAD em integrar conhecimentos heterogêneos ¹⁰, bem como torná-los disponíveis à sociedade.

Basicamente, os protocolos são mensagens trocadas entre os agentes, do tipo fluxos de *dados* e de *controles*. As mensagens do tipo dados são pertinentes diretamente ao problema em foco. As mensagens do tipo controle exercem um caráter coordenativo da cooperação. Em todas instâncias, as mensagens são trocadas pelos seus módulos de *comunicação* (ver figura 3.1). No projeto de agentes pode-se verificar a existência de um ou mais módulos. Um módulo corresponde há um dos “pontos de vista” com que o agente deseja interagir. O campo de comunicação regula (controla) o comportamento isolado do agente em relação à sociedade [Dem93]. Um exemplo prático da descrição acima são as redes de computadores, onde é necessário a presença de um esquema de protocolos na troca de dados entre duas máquinas.

Os esquemas de protocolos para comunicação entre agentes apresentam-se em diversos níveis de complexidade, permitindo, assim, um tipo de categorização. Há esquemas sofisticados em que o agente inicialmente informa seus parceiros da sua habilidade em resolver parte do problema ou, ainda, comunica uma necessidade individual [DB92]. Em [Cd94, Cd96], a comunicação entre agentes na IAD é categorizada sob as seguintes visões:

¹⁰Leia-se agentes heterogêneos.

1. *Sem comunicação*: nesse caso, o agente infere racionalmente os planos dos demais agentes, sem a necessidade de comunicar-se com eles. Os agentes devem efetuar um sofisticado processamento de raciocínio local para compensar essa falta de comunicação com os demais agentes e decidirem uma ação ou interação apropriada;
 2. *Uso de primitivas de comunicação*: esse é restrito a algum conjunto finito de sinais fixos com interpretações fixas. Em verdade, esse reduzido número de sinais tem finalidades, como evitar conflitos em processos sequenciais ou ainda coordenar atividades paralelas. Obviamente que pedidos, comandos e intenções complexas não podem ser expressas por esses sinais fixos. Desse modo, o processo cooperativo se torna limitado;
 3. *Passando planos*: nessa abordagem, um agente A1 comunica seu plano completo a um agente A2. Por outro lado, A2 comunica o seu plano completo a A1. A escolha do plano é a do primeiro que chegar. Contudo, há duas características sobre esse esquema. Primeiramente, a passagem de planos é de alto custo computacional e, em segundo, não há nenhuma garantia de que o plano comunicado seja autorizado pelo agente receptor. Críticas sobre a dificuldade dessa passagem completa de planos são: o meio de comunicação não é garantido quanto à fidelidade das mensagens emitidas e recebidas; em aplicações reais, esse esquema é inviável, devido à incerteza do estado corrente, bem como do estado futuro. Ou seja, os planos tornam-se ultrapassados muito rapidamente;
 4. *Trocando informações via “blackboard”*: em IA, o modelo mais comumente usado de memória global compartilhada é o “blackboard”. Os agentes escrevem mensagens, atualizam seus resultados parciais e encontram informações. Normalmente, o “blackboard” é particionado em vários níveis de abstração, segundo o problema nos quais os agentes interagem num nível em particular. O paradigma do “blackboard” representa o clássico problema do fluxo de informação ser *orientado a dados* ou *orientado a objetivos*. Isso o torna um dos esquemas mais utilizados na IAD;
 5. *Passando mensagens*: esse esquema é representado pelo paradigma de atores [RK94, Hew81]. Um ator é um tipo de agente computacional que age em resposta a um processamento de uma comunicação, no caso, uma mensagem. As ações desempenhadas são do tipo: enviar uma mensagem para si próprio ou para outros agentes, criar mais atores, especificar a substituição comportamental. Essa substituição determina um novo estado para o ator. A idéia original de atores é fundamentalmente interessante, quando o controle de SMA é especificado em termos das *estruturas de comunicação*. Nesse paradigma, os atores possuem estruturas simplificadas e o modelo é sem uma sintaxe formal, sem semântica ou pragmatismos para tais *estruturas de comunicação*. Conseqüentemente, esse modelo foi raramente usado como uma teoria de comunicação entre agentes;
-

6. *Comunicação de alto nível*: envolve áreas da pesquisa, como entendimento de linguagem natural (LN) e intenções na comunicação. Na IAD é relevante o fato das pesquisas se dirigirem a raciocínios de múltiplos agentes com distintos ou até mesmo estados intencionais contraditórios, como crenças, fatos e intenções. Numa comunicação de LN, mesmo para simples tipos de diálogos, há o uso do conceito de estados mentais, como crenças, fatos, intenções e bases experimentais de conhecimento. Como essa formulação, a compreensão de diálogos é factível mesmo com a ausência de algumas palavras, uma vez que se conhece as bases do diálogo e suas intenções. Em verdade, diálogos entre agentes permitem a geração e a interpretação de declarações (verbalizações, opiniões, discursos, asserções, etc), que são ações faladas e planejadas, definindo um estado mental para o emissor (crenças, comprometimentos e intenções). Esses estados consistem em induzir um estado mental particular do receptor. “*Através de um diálogo apropriado, os agentes podem convergir sobre planos mentais compartilhados (isto é, redes de crenças, comprometimentos, e intenções), pois dessa maneira eles coordenariam suas atividades*” [Cd94];
7. *Interação homem-máquina*: esse tipo crítico de comunicação valida a eficácia de uma aplicação na interação entre usuários e agentes artificiais. Em resumo, há uma atividade colaborativa, ou integradora, ou cooperativa entre sistemas computacionais e pessoas, possivelmente distribuídas em diferentes localizações. Os agentes artificiais podem atuar como assistentes inteligentes de seus usuários, que normalmente realizam tarefas tediosas ou que desperdiçam muito tempo, como pesquisas e obtenção de informações, sincronização de agendas de um grupo de pessoas, escalonamentos, etc.

Independente dos esquemas de comunicação mencionados acima, a interação entre agentes deve exibir um desempenho predizível e eficiente. Em uma visão de protocolos, eles restringem *o que, quando e como* se comunicar. Desse modo, os projetistas devem ter mente aspectos como [Ros94]: reprodução e distribuição de informações consistentes, estabelecimento de comprometimentos quanto à alocação de recursos [Tur91], acesso e execução de comandos remotos, consultas e processamento em bases de conhecimento, contratação e negociação de serviços entre os agentes, etc.

3.7.2 A Comunicação sob a Ótica deste Trabalho

Para fins do corrente trabalho, utiliza-se um esquema simplificado para comunicação entre agentes, próximo ao uso de primitivas. O esquema de primitivas confronta-se diretamente com a cooperação, pois há uma restrição sintática e semântica em suas entidades de comunicação. A cooperação, por outro lado, tem objetivos fundamentais dentro da IAD como [Cor89]:

- Aumento da taxa de realização de tarefas através do paralelismo;

- Aumento do conjunto ou do escopo de tarefas executáveis pelo compartilhamento de recursos (informação, esperteza, dispositivos físicos, etc);
- Aumento das chances de finalização das tarefas pelo engajamento redundante de agentes, possivelmente com diferentes métodos de resolução;
- Diminuição da interferência entre tarefas, evitando interações indesejáveis.

De modo complementar, M. Klein [Kle91] define situações cooperativas quando as partes são unidas por um objetivo “*super ordenado*”, para se atingir uma solução global. Isso implica em sacrifícios das partes menores, onde o objetivo global é dividido e a melhor solução surge da cooperação entre as partes.

Objetivando-se manter um *grau de cooperatividade*¹¹, o problema é contornado por uma rede de agentes com objetivos orientados ao problema, análoga à abordagem SDP. Não é uma rede controladora de agentes, pois isso levaria a uma enorme dificuldade de restringir e controlar uma comunicação apropriada dos agentes. Na construção dessas redes controladoras de multiagentes, E.H. Durfee e outros [Cor89] sugerem que elas apresentem um balanceamento efetivo entre o uso da comunicação e recursos requisitados, alta confiabilidade, boa sensibilidade das respostas diante de imprevistos e soluções de boa qualidade baseadas em critérios específicos.

Esses critérios da cooperação e das redes controladoras de agentes restringem-se a um contexto em que os próprios agentes formam redes dirigidas a objetivos cooperativos. Em resumo, *a abordagem desejada é semelhante a SDP com um esquema de comunicação por meio de primitivas*. No esquema de SDP aplicado a esta tese, alguns conceitos de IAD tornam-se específicos e serão discutidos a seguir.

3.7.3 Elementos e Conceitos

Os elementos e os conceitos nesta seção seguem a visão desta tese. Os agentes e suas mensagens são denominados de *elementos*. Por outro lado, a abstração resultante das conexões entre os agentes refere-se aos *conceitos*. Os agentes são primários e podem ser pensados como um sistema de transcrição por meio de regras, como exemplo, regras de produção. Os conceitos são oriundos dos acoplamentos e interações entre os agentes.

A apresentação é feita com uma linguagem gráfica, mostrando os tipos de interações entre agentes. A comunicação entre agentes ainda é um dos focos de pesquisa em aberto na IAD. Apenas as interações de interesse são aqui ilustradas. O enfoque dado sobre esses conceitos e elementos é o biológico [Zie92].

Na figura 3.2 encontram-se os mecanismos de *cooperação* (\otimes) e *competição* (\bullet) entre dois ou mais agentes. A cooperação ocorre quando dois ou mais agentes operam de modo paralelo e concorrente, com objetivo de promover uma ativação de um outro agente ou uma meta global. Num tratamento mais formal para cooperação tem-se:

¹¹É caracterizado pela quantidade de cooperação entre os agentes, que variam por serem totalmente cooperativos ou até antagônicos.

$$m_1 \otimes_1 m_2 \otimes_2 \dots \otimes_n m_n \xrightarrow{Ag_k} m_k$$

O operador \otimes_i corresponde ao grau de cooperatividade que um *agente_i* apresenta com o seu sinal ou mensagem m_i . A rigor, a fusão dessas cooperatividades ocorre no agente receptor, Ag_k , isto é, não de modo isolado como na figura 3.2. O operador “ \otimes ”, pode ocorrer pelo acoplamento direto ou por um comprometimento a priori, uma negociação prévia [Gas91]. Nesse último caso, são as chamadas *redes contratuais* [Gas91].

A competição está presente na maioria dos esquemas biológicos, exemplos em [MAA95]. O operador “ \bullet ” refere-se a um ou mais agentes que buscam maximizar seus potenciais locais, sob critérios e recursos limitados de uma mesma fonte. Nesse caso, o operador “ \bullet ” (competição) encontra-se associado as entradas de dois ou mais

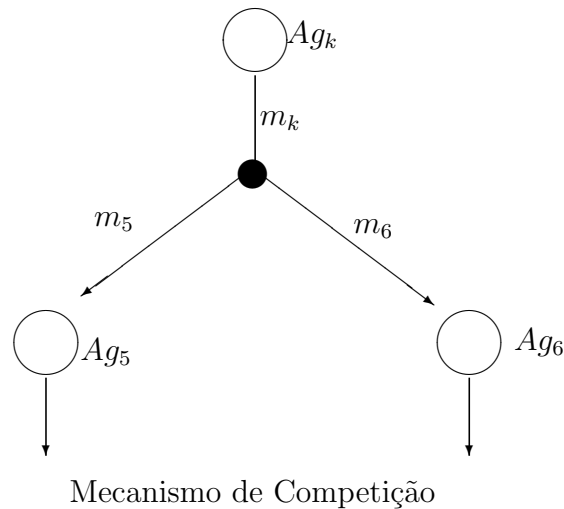
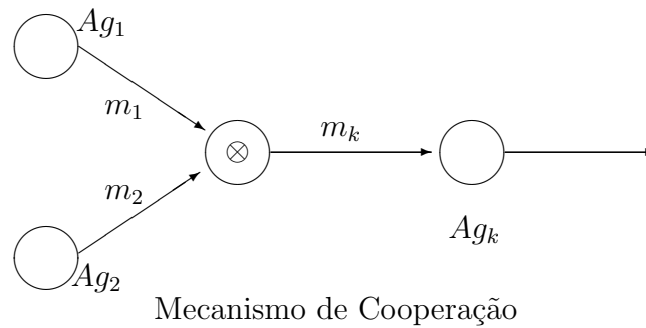


Figura 3.2: Esquemas de cooperação e competição

agentes, embora seus objetivos sejam concorrentes e distintos. Uma formalização é dada por:

$$m_k \xrightarrow{Ag_k} m_1 \bullet_1 m_2 \bullet_2 \dots \bullet_n m_n$$

Aqui, um agente Ag_i com um potencial de competitividade (barganha) \bullet_i espera receber uma mensagem de entrada m_i , na disputa com outros agentes sobre a disponibilidade de recursos em Ag_k . Em redes neuronais, essa propriedade pode ser exibida pela filtragem do sinal axonal. O sinal de saída no axônio é m_k , mas de acordo com as afinidades dos neurotransmissores junto aos dendritos, diferentes sinais m_1 , m_2 , ..., m_n , podem ser obtidos nas entradas [Roc92]. Essa afirmação é reforçada pela topologia neural feita para redes *convergentes* e *divergentes* em [VO92], que apresentam definições formais análogas às realizadas para cooperação e competição, respectivamente.

Outros conceitos inerentes à IAD (e do mundo real) são a *limitação de recursos* [Tur91] e o *conflito*. O conflito entre agentes ocorre devido a uma incompatibilidade de objetivo ou de coerência global. Segundo M. Klein [Kle91], em situações de conflito competitivos, cada agente tem em mente o seu benefício próprio. Não há nenhum interesse em atingir uma solução global ótima ou próxima a ela. Em suma, os interesses são particulares aos agentes e distantes de uma solução global.

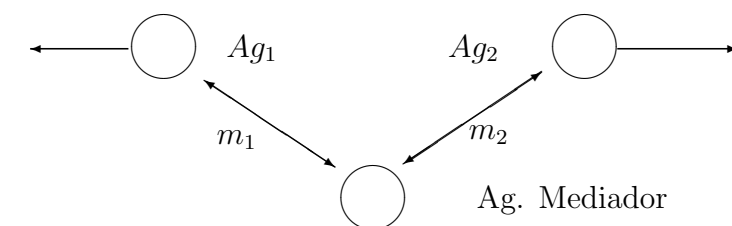
Da teoria dos conjuntos, o conceito conflito pode ser ilustrado aproximadamente por: dada uma informação (um conhecimento) x sobre um sistema fechado e \bar{x} como seu complemento, leva-se as seguintes premissas:

$$x + \bar{x} \begin{cases} = 1 \Rightarrow \text{condição ideal} \\ < 1 \Rightarrow \text{falta de evidência} \\ > 1 \Rightarrow \text{conflito} \end{cases}$$

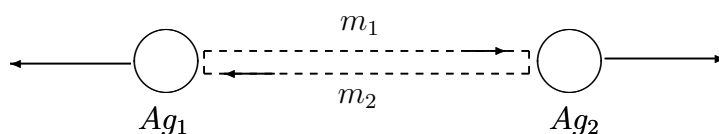
Embora o conflito represente um impasse sistêmico, ele pode ser visualizado como um processo de aprendizagem entre os agentes. Sob essa hipótese, consideram-se os benefícios provocados por um conflito. Na detecção de um conflito, dispara-se um processo de resolução, chamado de *negociação*. Necessita-se de um *mecanismo de negociação* que objetive resolver o impasse existente entre os agentes.

Dois processos de negociação são ilustrados na figura 3.3. O primeiro necessita de um agente-juiz (um árbitro) para efetuar uma ação decisória entre os agentes. De acordo com a figura, o conflito é detectado e resolvido pelo árbitro, uma vez que os agentes Ag_1 e Ag_2 não se comunicam. De modo oportuno, é apresentado no capítulo 5 uma rede de agentes interconectados, que se encontram situacionalmente em conflito. A solução adotada foi a negociação por meio de um árbitro, que promova certas prioridades na rede de agentes.

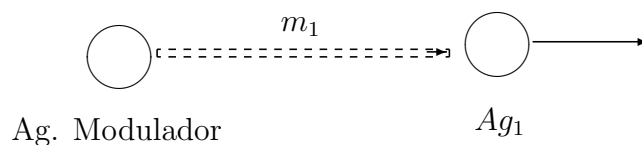
Num segundo caso, há um processo de *decisão consensual* entre os agentes, sem a presença de um agente-juiz. Nesse exemplo, a habilidade cognitiva interna dos agentes é sofisticada, pois envolve mecanismos complexos típicos da comunicação social humana. Esses agentes cognitivos não pertencem ao escopo deste trabalho.



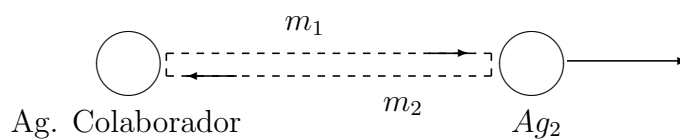
Negociação com árbitro



Negociação de comum acordo



Modulação



Colaboração requisitada

Figura 3.3: Outros conceitos em IAD

Num terceiro exemplo, um agente detector de conflito pode agir de forma autoritária e assíncrona sobre os agentes de níveis inferiores. Essa atuação pode ocorrer em vários níveis de uma rede de agentes. O esquema é próximo ao utilizado em arquiteturas de *englobamento*, proposto por R. Brooks [Bro91c]. Esse agente é chamado de *modulador* pelo fato de enviar uma mensagem m_1 com diversos níveis de intensidade. Análogo a uma saída axonal, a intensidade e a frequência dos pulsos neurais emitidos podem atuar de modo inibidor ou não. Em resumo, a *modulação*¹² é uma ação com influência por parte de um agente sobre outro. Independentemente da requisição de um agente solicitante, um agente modulador encontra-se *situado* [Mae91c]

¹²Termo emprestado da biologia.

de modo a interferir em Ag_1 (ver figura 3.3). Nesse caso, há uma comunicação imposta e não protocolar. Adicionalmente, essa imposição pode ser do tipo implícita ou explícita. A *colaboração* entre dois agentes ocorre quando Ag_2 faz um pedido de ajuda a um agente colaborador (ver figura 3.3). Nesse caso, trocam-se os dados e o controle entre os agentes. Observa-se que a colaboração é um conceito distinto da co-operação. O agente colaborador não tem um papel ativo sobre o agente requisitante ou da solução global.

Capítulo 4

Lógica Nebulosa

4.1 Introdução

Tendo em vista que a IA trata de problemas complexos, tanto em tamanho como em nível de detalhes, imprecisões de dados e informações ocorrem de forma intrínseca. Essas imprecisões precisam ser tratadas de modo a recompor um valor original ou propagar um valor confiável. Esse valor deve ser suficientemente representativo de modo a retratar um estado corrente do sistema.

O segmento em paralelo a IA, que trata desses assuntos, é a área de *Cálculo de Incertezas*. Essa área abriga muitas técnicas com diferentes paradigmas de abordagem. Entre as principais visões do cálculo de incertezas em IA, citam-se a *nebulosa*, a *probabilística*, a *possibilista* e a de *crenças*. Essas visões são também chamadas de *modelos*, pois tipos específicos de informações são ajustados e tratados sobre esses modelos. Uma introdução sobre o assunto é encontrada nas referências [Bit96, RK94].

Neste capítulo, a visão nebulosa apresenta um interesse contextual em particular. Em um dos níveis da arquitetura proposta, os agentes modelados são do tipo nebuloso ou “fuzzy”. A visão nebulosa é apresentada pela lógica nebulosa, que objetiva realizar raciocínios aproximados. Esse raciocínio é feito por um esquema inferencial indutivo, a partir de um conhecimento impreciso. Seu embasamento teórico assenta-se sobre os *conjuntos nebulosos*. A principal característica desses conjuntos é a indefinição de suas fronteiras. Quanto a lógica, o encaminhamento é a partir do paradigma dos sistemas de produção. Desse modo, tem-se, por meio da lógica nebulosa, um esquema de representação de conhecimento aliado a estratégias de raciocínios aproximados.

4.2 Resumo Histórico

No início dos anos setenta, pesquisadores da IA concentraram seus esforços em tornar as lógicas em geral aplicáveis à resolução de problemas. Pesquisadores como Nils Nillson [Win87] chegaram a abordar a IA como uma questão de lógica aplicada.

Inicialmente, a lógica clássica ou de dois valores teve suas raízes em Aristóteles, há aproximadamente 2500 anos. Tratando-se de um filósofo, Aristóteles discutia a lógica

com base em uma *interpretação*, cuja problemática era demonstrar a manutenção de uma verdade em um determinado contexto. Esse questionamento leva à análise de eventos futuros, cujos valores podem se tornar em verdades ou falsidades, potencialmente em ambos ou em nenhum deles. Os conceitos de verdade (V) e falso (F) foram investigados durante séculos por filósofos, matemáticos e lingüistas, mas sem os formalismos alcançados no final do século passado, por Frege¹, Peano², B. Russel, A.N. Whitehead³ e outros [KK68].

Um sistema lógico consiste em um conjunto de *fórmulas* e em um *conjunto de regras de inferência*. As fórmulas são sentenças pertencentes a uma linguagem formal cuja sintaxe é conhecida. A montagem de uma fórmula deve seguir uma sintaxe pertinente. Cada fórmula pode ser associada a um valor verdade. Esses valores são Verdadeiro (V) ou Falso (F). A parte da lógica que estuda regras para obtenção dos valores verdade é chamada de *Teoria dos Modelos*.

Um maneira formal de realizar essa associação entre fórmulas e valores verdade foi proposta por Tarski (1936, 1956). A questão básica é como associar os valores V e F às fórmulas. Dois problemas importantes estão relacionados com o valor verdade de uma fórmula: os problemas da *validade* e da *tautologia*. A validade refere-se à possibilidade de uma fórmula dada apresentar o valor verdade verdadeiro. A tautologia mostra se uma fórmula é ou não sempre verdadeira, tornando-se um *teorema lógico* se isto for provado.

Uma regra de inferência é uma regra sintática, que quando aplicada repetidamente a uma fórmula válida, gera apenas novas fórmulas válidas. As regras de inferência fornecem uma estrutura dedutiva à linguagem lógica. A seqüência de fórmulas geradas por meio da aplicação de regras de inferência sobre uma fórmula inicial dada é chamada de *Prova*. A parte da lógica que estuda as provas é chamada de *Teoria das Provas*. Dois importantes resultados associam a teoria dos modelos à teoria das provas. O primeiro é um resultado negativo e é devido a Church, que afirma não existir um método geral capaz de decidir, em um número finito de passos, se uma fórmula é válida. O segundo resultado é positivo e é devido a Gödel. Ele afirma que existem sistemas lógicos de prova nos quais toda e qualquer fórmula válida pode ser provada. Essa propriedade de um sistema de prova é chamada de *completude* do sistema. Seus resultados são a base do poder da lógica como método aplicado a IA e também de sua fraqueza. O teorema de Gödel fornece um método para a inferência de sentenças, cujo sentido é formalmente definido pela semântica de Tarski. No entanto, o teorema de Church afirma que os métodos para a verificação da validade de fórmulas são inerentemente ineficientes e que podem até nunca terminar.

¹Publicou em 1879 o “*Begriffsschrift*”, um dos primeiros tratados da lógica formal.

²Publicou em 1894 o “*Notations de Logique Mathématique*”, que introduziu uma das melhores notação à lógica formal.

³Whitehead e Russel publicaram em 1910 o “*Principia Mathematica*”, reeditado em 1925, um outro tratado à lógica discordando de algumas idéias de Frege.

4.3 A Lógica Nebulosa

Além do uso de fórmulas predicativas na lógica binária⁴ iniciada por Aristóteles, e melhor entendida apenas neste século, outras lógicas também foram propostas. Lógicas com propósitos e diversos universos de discursos. Algumas visando suas refutações em seu próprio domínio, a exemplo da lógica de circunscrição proposta por John McCarthy, outras buscando suporte em outros mundos lógicos, como a lógica modal em um de seus esquemas de prova.

Mais recentemente, uma lógica chamada de “nebulosa” foi apresentada por Lotfi A. Zadeh, em 1965 [Kos94, Rag79, Dub79]. A origem dessa lógica é atribuída a Zadeh por ter sido o primeiro a usar o termo “*fuzzy logic*”, que se popularizou por seus conceitos intuitivos na modelagem do conhecimento, com um embasamento matemático relativamente simples e flexível. Apesar do tratamento matemático ser simples, ele é igualmente robusto. A lógica nebulosa tem sido usada em larga escala em aplicações industriais, com excelentes resultados [Kos94]. Atualmente, tornou-se uma técnica não exclusiva da IA, com reputação incontestável em vários segmentos da ciência e da indústria [TS87]. A atenção das diversas áreas sobre essa teoria se deve a sua eficiência e naturalidade para descrever conjuntos aproximados, vagos, nebulosos, mal definidos ou imprecisos. E, finalmente, permite considerações de detalhes relativos às imperfeições intrínsecas aos sistemas do mundo real.

Dois acontecimentos deste século motivaram os primeiros passos em direção à lógica nebulosa ou vaga, apresentada por Zadeh em 1965. O primeiro deles foi dado por Bertrand Russel, que redescobriu o paradoxo dos gregos quanto à fundamentação da matemática moderna: a *bivalência*. Um paradoxo matemático é conhecido por *auto-referência*, onde as proposições se afirmam e se negam a si próprias, tudo terminando em “ A and \bar{A} ”. Exemplos clássicos são: “*Um cretense disse que todos os cretenses são mentirosos*”. “*O homem que barbeia todos e somente aqueles que não barbeiam a si próprio; e o barbeiro barbeia a si próprio*”? De um lado da rua: “*A sentença do outro lado da rua é verdade*”, do outro lado da rua: “*A sentença do outro lado da rua é falsa*”. Russel descobriu “*o conjunto de todos os conjuntos que não são membros de si mesmos*” [Kos94]. Por essas indefinições, Bertrand Russel questionou e colocou um “fim” na certeza matemática de “ V ” ou “ F ”, existente por 2500 anos.

O segundo acontecimento foi dado por Werner Heisenberg com o *princípio da incerteza da mecânica quântica*. Heisenberg provou que em mecânica quântica algumas coisas nunca poderiam ser conhecidas, pois há uma incerteza nos átomos que compõem nossas mentes. Mostrou que, mesmo na física, a verdade seria um problema de grau, pertinência, ponderação etc. A idéia era de que quanto mais próximo se olha a matéria, menos se vê. Desse modo, mesmo com uma informação total, não se asseguraria alguma coisa com 100% de certeza. Bart Kosko [Kos94] (capítulos 6 a 8) argumenta que esses dois eventos científicos nas décadas de 20 e 30, dados por B. Russel e W. Heisenberg, foram os passos iniciais em direção à lógica nebulosa.

⁴Cálculo de predicados ou lógica de 1a. ordem.

Na mesma época, um importante trabalho foi produzido pelo polonês Jan Łukasiewicz, em torno da *lógica multivalorada* ou de *n-valores*. Esta é uma generalização da lógica de três valores de Bochvar [Kli88]. A idéia é que possa existir um número arbitrário de valores verdade. Os valores verdades são normalmente rotulados pelos números racionais no intervalo $[0, 1]$. Esses valores são obtidos pela divisão em intervalos iguais no domínio dos valores entre $[0, 1]$. O conjunto de valores verdade T_n de uma lógica de n-valores (onde $n \geq 2$) é então definida por:

$$T_n = \{0 = \frac{0}{n-1}, \frac{1}{n-1}, \frac{2}{n-1}, \dots, \frac{n-2}{n-1}, \frac{n-1}{n-1} = 1\}$$

Esses valores podem ser interpretados como *graus de verdade* [Kli88]. A partir dos valores definidos por T_n , Łukasiewicz define algumas operações primitivas para esse conjunto multivalorado, dadas por:

$$\bar{a} = 1 - a$$

$$a \wedge b = \min(a, b)$$

$$a \vee b = \max(a, b)$$

$$a \Rightarrow b = \min(1, 1 + b - a)$$

$$a \Leftrightarrow b = 1 - |a - b|$$

Na literatura, os termos T_n freqüentemente são definidos por L_n , em lembrança a Łukasiewicz. Para o caso de $n = 2$, essa lógica se reduz à lógica clássica ou de dois valores. Para o caso extremo de $n = \infty$, tem-se uma lógica infinitamente valorada (L_∞), cujos valores verdades são os números racionais no intervalo unitário $[0, 1]$. A menos de uma exceção, essa lógica infinitamente valorada indica os valores verdades que são representados por todos números reais entre $[0, 1]$. Tratando-se de valores reais, é também chamada de *lógica padrão de Łukasiewicz* ou L_1 . O uso do subscrito “1” representa a cardinalidade de um *continuum*.

Dado o isomorfismo existente entre a lógica e a teoria dos conjuntos, observa-se que a L_1 de Łukasiewicz é isomórfica aos fundamentos de uma teoria de conjuntos nebulosos ou vagos⁵. Essa teoria trata dos conjuntos cujas fronteiras não estão bem definidas. Desse modo, a *pertinência* de um elemento também é imprecisa [Zad94]. Várias propostas quanto aos limites superiores e inferiores dos conjuntos nebulosos podem ser encontradas em [Kli88]. Em L_1 foi criado um conjunto de operadores: \min , \max e $1 - a$, que são equivalentes à interseção, união e complemento, respectivamente. Raciocínio análogo às equivalências existentes entre a lógica de dois valores e a teoria dos conjuntos de Euclides.

⁵Alguns conceitos básicos são apresentados ao longo do texto, detalhes em [Kli88], seção 1.4

4.3.1 Teoria dos Conjuntos Nebulosos

Nesta subseção, resumem-se os conceitos básicos da teoria dos conjuntos nebulosos, tendo por base os conceitos de *lógica multivalorada de Lukasiewicz*. Inicialmente, há o conceito da pertinência de um elemento em um dado conjunto. Em conjuntos nebulosos, todos elementos possuem um *grau de pertinência*, dado por:

$$\mu_A(x) \rightarrow [0, 1]$$

onde $[0, 1]$ define o intervalo dos números reais entre 0 a 1 inclusive. Na teoria dos conjuntos tradicional, essa função ou grau de pertinência assumiria dois valores: 0 ou 1. Isso é equivalente a “ x ” pertencer ao conjunto “ A ” ou não. Uma boa maneira de visualizar esse conceito é observar a nuvem do gradiente da explosão de uma estrela nova.

Dado que todo elemento x_i possui uma pertinência $\mu_A(x_i)$ com o conjunto suporte A , esse pode ser obtido pela soma de todos os x_i , portanto:

$$A = \mu_1/x_1 + \mu_2/x_2 + \mu_3/x_3 + \dots \mu_n/x_n$$

O símbolo “/” é usado para associar cada elemento do conjunto suporte A a seu valor de pertinência. O símbolo “+” significa algum tipo de adição algébrica de elementos, uma vez que se tem um par de valores para cada elemento de A . Caso A seja finito e contável, essa expressão pode ser reescrita em:

$$A = \sum_{i=1}^n \mu_i/x_i$$

Analogamente, quando x é definido sobre o intervalo dos números reais, essa fórmula é reescrita em:

$$A = \int_X \mu_A(x)/x$$

Por definição, a *altura* de um conjunto nebuloso é o maior grau de pertinência obtido por qualquer um dos elementos do conjunto. Caso pelo menos um dos elementos atinja o valor máximo do intervalo de pertinência (por exemplo, entre 0 e 1), o conjunto é definido como sendo *normalizado* [Kli88]. O interesse prático reside sobre conjuntos normalizados, onde pelo menos um termo atinja o valor de verdade absoluta sob alguma circunstância.

As operações básicas da função de pertinência para complemento, interseção e união [Kli88] são definidas por:

$$\mu_{A \cup B}(x) = \max[\mu_A(x), \mu_B(x)] \quad (4.1)$$

$$\mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)] \quad (4.2)$$

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad (4.3)$$

O operador *máx* opera sobre cada elemento x tal que $x \in (A \cup B)$ na união, e *mín* opera sobre cada elemento x tal que $x \in (A \cap B)$ na interseção. No caso da união, cada x presente em $A \cup B$, $\mu_{A \cup B}(x)$ vai ser definido pelo valor de maior pertinência em A ou B . No caso da interseção, $\mu_{A \cap B}(x)$ vai representar o menor valor de x presente em $A \cap B$, de modo que x pertença a A e B .

4.4 Operadores da Lógica Nebulosa

Para maioria das lógicas, há um universo de discurso (domínios) onde são definidas as proposições, uma sintaxe e um conjunto de regras para manipulações inferenciais. Um dos pontos que se questiona é como realizar associações entre os valores de entrada e saída em um dado domínio. O que faz um determinado valor no precedente, levar a um outro valor no conseqüente? Um dos encaminhamentos possíveis é pela equivalência entre a lógica clássica e regras de produção. Há um conjunto de assertivas A_i , que implicam numa conclusão B , como: $A_1 \wedge A_2 \dots \wedge A_n \rightarrow B$. Essa questão, C.C. Lee chamou de “*função de implicação nebulosa*” [Lee90b]. Em resumo, tem-se um domínio de entrada que é relacionado a um domínio de saída. A pergunta é: “*como realizar operações entre essas entradas e saídas?*” Uma das respostas é por meio de operadores de domínios!

A operação entre os domínios pode ser entendida como operações de um grupo algébrico. A função implicação ou inferência (ou conseqüência) resulta num fechamento sobre os domínios da entrada e saída. Nesse momento, L. Zadeh é o pioneiro no sentido de associar uma teoria de conjuntos nebulosos a uma lógica. Desse modo, os conhecimentos são modelados por proposições e regras imprecisas. A sintaxe é próxima ao cálculo de primeira ordem (assunto das seções seguintes). Quanto à inferência, é desejável que atenda às características sugeridas por Baldwin e Pilsworth [Lee90b]:

- Possua propriedades fundamentais (definidas nas subseções seguintes);
- Possua propriedade de suavidade (“*smoothness*”);
- Possua inferência irrestrita;
- Possua simetria generalizada quanto a *modus ponens* e *modus tollens*;
- E, finalmente, que possua uma medida de propagação de nebulosidade.

Objetivando-se o atendimento aos itens acima, existe uma necessidade de uma fundamentação matemática, que defina como os elementos de um conjunto se relacionam. Essa matemática é calculada com operações básicas sobre termos antecedentes ou precedentes, conseqüentes e, finalmente, como se relacionam numa regra de produção nebulosa. Dois operadores básicos relacionam diretamente a união e interseção de conjuntos nebulosos, conhecidos por *t-norma* (\circ) para interseção e *t-conorma* (\square) para união. As propostas citadas anteriormente *máx* e *mín* são casos particulares destes grupos. Suas definições formais são dadas por:

4.4.1 Operador t-norma (\circ):

O operador t-norma ou \circ é definido sobre $[0, 1] \times [0, 1] \rightarrow [0, 1]$, de forma que as seguintes propriedades sejam seguidas:

- a) $a \circ 1 = a$ (onde 1 é o elemento neutro)
- b) $a \circ b = b \circ a$ (comutatividade)
- c) se $a \geq c, b \geq d$ então $a \circ b \geq c$ (monotonicidade)
- d) $a \circ b \circ c = a \circ (b \circ c) = (a \circ b) \circ c$ (associatividade)

Generalizando, tem-se:

$$a_1 \circ a_2 \circ \dots \circ a_n = \circ_{i=1}^n a_i$$

Caso particular é o operador *min* :: ou :: (\wedge):

$$a_1 \wedge a_2 \wedge \dots \wedge a_n = \bigwedge_{i=1}^n a_i$$

ou ainda $a \wedge b = \min(a, b)$

4.4.2 Operador s-norma ou t-conorma (\square):

O operador t-conorma ou \square é definido sobre $[0, 1] \times [0, 1] \rightarrow [0, 1]$, de forma que as seguintes restrições sejam seguidas:

- a) $a \square 0 = a$ (onde 0 é o elemento neutro)
- b) as propriedades b) a d) das propriedades de t-norma (\circ).

Generalizando, tem-se:

$$a_1 \square a_2 \square \dots \square a_n = \square_{i=1}^n a_i$$

Caso particular é o operador *máx* :: ou :: (\vee):

$$a_1 \vee a_2 \vee \dots \vee a_n = \bigvee_{i=1}^n a_i$$

ou ainda $a \vee b = \max(a, b)$

Os operadores t-norma e t-conorma generalizam as operações “and” e “or”, respectivamente na lógica de multivalores [Kac87]. Esses operadores apresentam um isomorfismo com a interseção e união, respectivamente, na teoria dos conjuntos. O complemento é igualmente dado pela expressão 4.3. Onde A representa um conjunto de valores e sua negação é dada por \overline{A} .

4.4.3 Operadores Variantes

Os operadores nebulosos “ \circ ” e “ \square ” podem apresentar variações na associação de dois ou mais elementos em um conjunto nebuloso. Essas variações foram e continuam sendo sugeridas por vários autores [Kli88]. Algumas dessas possibilidades de variações são apresentadas na tabela 4.1.

Esses operadores têm propósitos diversos. O operador mais comumente utilizado é o de Zadeh. Esse operador explora os limites superiores e inferiores de uma operação nebulosa entre dois ou mais conjuntos. Quanto à negação, normalmente, utiliza-se o seguinte complemento:

Referência	\circ (t-norma)	\square (t-conorma)	Parâmetros
Zadeh	$\min(x, y)$	$\max(x, y)$	
Probabilista	$x \cdot y$	$x + y - xy$	
Lukasiewicz	$\max(x + y - 1, 0)$	$\min(x + y, 1)$	
Hamacher	$\frac{xy}{\gamma + (1-\gamma)(x+y-xy)}$	$\frac{x+y-(2-\gamma)xy}{1-(1-\gamma)xy}$	$\gamma \in (0, \infty)$
Dubois&Prade	$\frac{xy}{\max(x, y, \alpha)}$	$\frac{x+y-xy-\min(x, y, 1-\alpha)}{\max(1-x, 1-y, \alpha)}$	$\alpha \in (0, 1)$
Yager	$1 - \min[1, ((1-x)^w + (1-y)^w)^{1/w}]$	$\min[1, (x^w + y^w)^{1/w}]$	$w \in (0, \infty)$

Tabela 4.1: Operadores Duais t-norma e t-conorma

$$\bar{x} = 1 - x$$

Quanto à implicação, serve para avaliar um grau de confirmabilidade dos antecedentes de uma regra de produção nebulosa. Dos vários termos antecedentes de uma regra do tipo:

$$(x_1 \text{ é } A_{1j}) \wedge (x_2 \text{ é } A_{2k}) \wedge \dots \wedge (x_n \text{ é } A_{nn}) \rightarrow (y \text{ é } B)$$

a questão é propagar e relacionar essa certeza com o termo conseqüente B da regra. Isso é, conhecer o grau de pertinência final do termo B . Novamente, várias são as propostas. Ver tabela 4.2.

Referências	Implicação
Kleene	$\max(1 - x, y)$
Lukasiewicz	$\min(1 - x + y, 1)$
Gödel	1 se $(x \leq y)$ e y se $(x > y)$
Mandani	$\min(x, y)$
Larsen	$x \cdot y$

Tabela 4.2: Principais Operadores de Implicação

A tabela 4.2 foi reportada de [Bit96]. O operador “.” é definido como sendo o produto algébrico, dado por xy [Lee90a, Lee90b]. Novamente, esses operadores são característicos de aplicações específicas. Contudo, o operador de Mandani é o mais comumente usado. O operador de Mandani é essencialmente pessimista, pois escolhe o menor valor para a regra. É interessante em casos em que não haja uma certeza absoluta na parte antecedente da regra. Uma boa discussão sobre as escolhas por Zadeh e Mandani [Mam85] é encontrado em [Miz95].

4.5 Conhecimento Impreciso

Esta seção motiva o uso da lógica nebulosa como um refinamento do paradigma de sistemas baseado em regras de produção. Esse passo visa à representação do conhecimento e de um esquema de raciocínio usando inferências geradas a partir de regras nebulosas. Consideremos os seguintes exemplos:

Exemplo 1: O conjunto das pessoas nascidas em um estado. É um conjunto bem definido e sem ambigüidades em relação aos seus elementos e fronteiras. Todo cidadão tem uma certidão de nascimento onde consta o seu estado de origem.

Exemplo 2: O conjunto das pessoas satisfeitas com seus salários em uma grande empresa.

No exemplo 1, tem-se um conjunto bem comportado. Contudo, no exemplo 2, a resposta é vaga, imprecisa, nebulosa, ... etc. Como alternativa estratégica, sugere-se associar um valor entre 0 e 1 a cada elemento desse conjunto. A idéia é qualificar a sua *pertinência* em torno de um valor nominal (de um péssimo a um ótimo salário), representativo ao conjunto dos salários. No caso, U é o conjunto de todos habitantes do Brasil e x um elemento, dado que $x \in U$. Essa função ou grau de pertinência é definida por $\mu_A(x)$ com respeito a um conjunto $A \subseteq U$ (A : conjunto de habitantes de um estado), isto é:

$$\mu_{A(x)} = \begin{cases} 1 & \text{se } x \in A \\ 0 & \text{se } x \notin A \end{cases} \quad (4.4)$$

A função $\mu_A(x) : U \rightarrow \{0, 1\}$ é chamada de *função característica* na teoria clássica de conjuntos.

Quanto ao exemplo 2, há uma variável implícita chamada de *salário*, que precisa ser quantificada e qualificada. Ou seja, necessita-se traçar uma *função descritora* ou de *pertinência* para definir os elementos desse conjunto e suas vizinhanças. Para esse caso, a *função de pertinência* $\mu_A(x)$ é dada por:

$$\mu_A(x) : U \rightarrow [0, 1]$$

O conjunto suporte de um conjunto nebuloso A é o subconjunto dos pontos x em U , de modo que $\mu_A(x) > 0$. O conjunto A é chamado *normalizado* se $\mu_A(x_i) = 1$, em pelo menos um x_i .

Contudo, o especialista necessita mapear o conhecimento sobre a *variável* salário, também conhecida como *variável lingüística*. Essa variável necessita ser categorizada, isto é, situar-se em subconjuntos característicos. Os elementos assemelhados são definidos por *termos*. Os termos concentram um conjunto de elementos de mesmas características. Zadeh denomina-a de “*compressão de dados*” ou *granularidade* do conjunto nebuloso [Zad94]. Exemplificando, o caso da variável *salário* (S) pode possuir os seguintes termos:

$$S = \{s_0, s_1, s_2, s_3\}$$

cuja semântica expressa uma faixa de valores, que possuem limites superior e inferior, e sendo normalizados possuem uma “*faixa de certeza*”. Por exemplo, os termos podem ser representados por curvas trapezoidais (ver tabela 4.3):

		a	b	c	d
s_0 :	“muito baixo”	100	100	150	200
s_1 :	“baixo”	150	200	300	500
s_2 :	“médio”	300	500	600	700
s_3 :	“bom”	600	700	1000	1000

Tabela 4.3: Curvas dos Termos da Variável Salário

A definição é sobre uma faixa de valores aproximados, fornecida a partir de um analista de salários. Nesse exemplo, o especialista mapeou os salários pagos pela empresa a partir de 100 a 1000, ponderando-os em faixas. Os pontos “ a ” e “ d ” representam os limites inferior e superior de cada termo, respectivamente. A faixa definida entre os pontos “ b ” e “ c ” é denominada como “*faixa de certeza*”. Ela reflete a segurança em afirmar que os valores nesse intervalo são membros de um dado termo. Obviamente que se $b = c$ tem-se uma curva triangular. O conhecimento sobre a variável nebulosa S pode ser expressa em curvas de conhecimento, do tipo da figura 4.1.

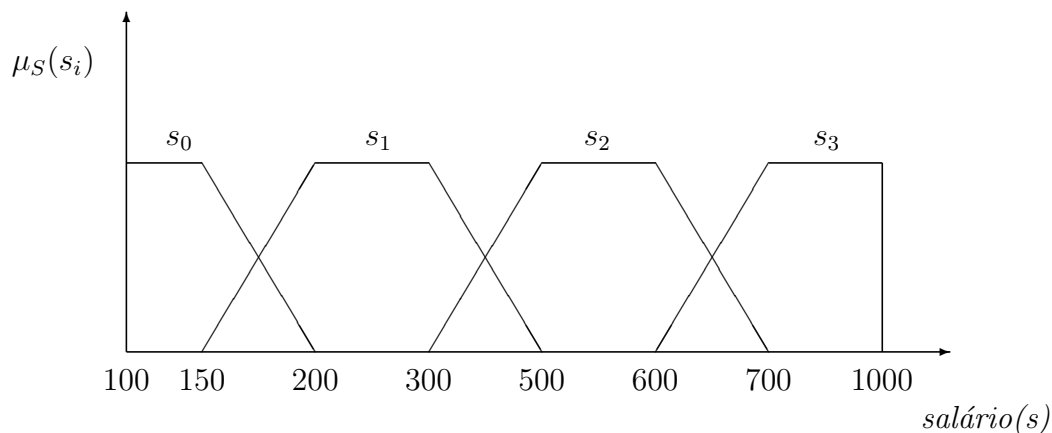


Figura 4.1: Descrição da variável salário

4.6 Regras de Produção Nebulosa

A partir da lógica nebulosa aqui enfocada, a abordagem que segue é dirigida a regras de produção. A partir delas, aplica-se um esquema de raciocínio aproximado [Yag84].

Esse esquema usa um método inferencial lógico, no qual o ponto de partida é um conhecimento impreciso, modelado a priori. A sintaxe de uma regra nebulosa é do tipo “*se-então*”. A aplicação prática dessa regra concentra-se na área de controle processos [Kos94, FG94]. Seu padrão para uma variável com um termo antecedente e um conseqüente na regra é dado por:

$$\text{se } (X \text{ é } A_i) \text{ então } (Y \text{ é } B_j)$$

para $i = 1, 2, 3, \dots, m$ e $j = 1, 2, 3, \dots, n$. O conjunto de regras que define esse par de variáveis é dado por $A_i \times B_j$ (produto cartesiano dos termos). A equação acima pode ser generalizada por:

$$\text{se } (X_1 \text{ é } A_i) \text{ e } (X_2 \text{ é } A_{i+1}) \text{ e } \dots \text{ e } (X_m \text{ é } A_{i+m}) \text{ então } (Y \text{ é } B_j)$$

É interpretado como uma restrição conjunta sobre os termos A e B, definido por:

$$(X, Y) \text{ é } A_i \times B_j \quad (4.5)$$

O conjunto de pares formado por $(X, Y) \text{ é } A_i \times B_j$ é conhecido como uma *relação nebulosa* f^* (“*f estrela*”). A equação 4.5 possui uma equivalência gráfica. Como exemplo, para duas variáveis nebulosas, essa equivalência é representada pela figura 4.2. As regiões ou áreas formadas pela interseção das curvas desses termos definem uma pertinência da regra em uma determinada faixa de valores. Essas áreas (ver figura 4.2) descrevem a “*suavidade*” com que as partes antecedente e conseqüente das regras se relacionam. Por exemplo, em A_1 há uma certeza ($\mu_V(x) = 1$) para regra $t_11 \rightarrow t_21$. Em A_4 , a regra deve ponderar dois termos antecedentes com seus respectivos μ para que a regra seja ponderada. As curvas dos termos são inclinadas e desse modo um cálculo dos μ deve ser levado em conta. Essa regra é dada por:

$$t_11 \text{ e } t_12 \rightarrow t_21$$

Em resumo, as áreas definem “*retalhos*” do conhecimento lingüístico, expresso sob a forma de regras nebulosas. Aos “*retalhos*” ou uma parte comprimida da variável, L. Zadeh chamou de *granularidade* das regras [Zad94]. Quanto maior o número de retalhos, maior é a imprecisão do conhecimento e maior é o número de regras. Tendo como efeito direto um maior número de cálculos.

Cada termo da regra nebulosa possui uma pertinência, que representa um *grau de confirmabilidade* [Adl86] do termo na regra. Supondo que a variável “*salário*” (S) esteja associada a uma variável chamada “*motivação pessoal*” (M), uma regra típica é dada por:

$$\text{se } (S \text{ é } \textit{baixo})_{\mu_S(s)} \text{ então } (M \text{ é } \textit{pouca})_{\mu_M(m)} \quad (4.6)$$

Onde $\mu_S(s)$ e $\mu_M(m)$ correspondem à pertinência dos termos “*baixo*” e “*pouca*”, respectivamente. Como já comentado, um conjunto de regras nebulosas com essa sintaxe expressa um conhecimento aproximado do especialista. A imprecisão dos termos

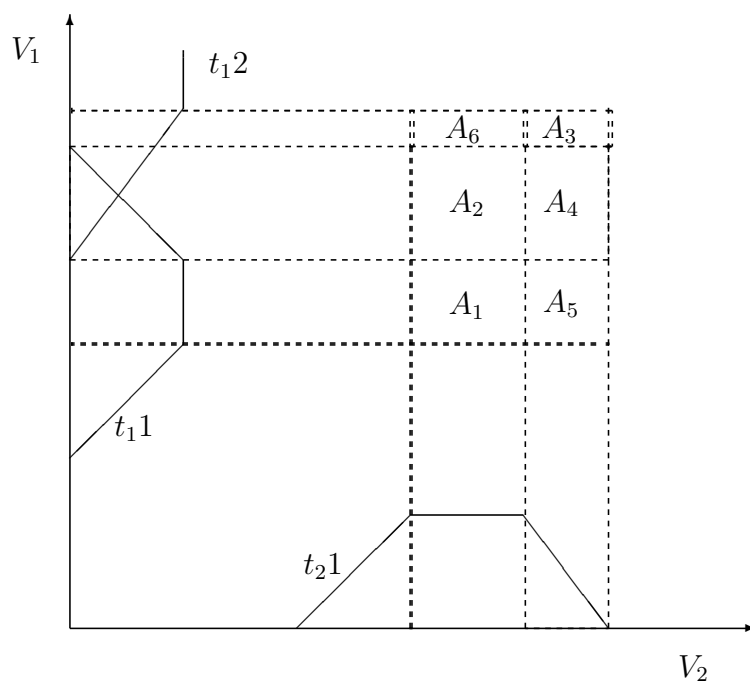


Figura 4.2: Curva granular para duas variáveis V_1 (antecedente) e V_2 (consequente)

é caracterizada por seus respectivos *graus de pertinência* $\mu_A(x)$, que são calculados a partir das curvas dos termos (ver figura 4.1).

4.7 Cálculo de Inferência

Um processo de raciocínio sobre regras nebulosas do tipo “*se-então*” é definido basicamente em quatro passos. O procedimento de cada passo pode ser individualizado segundo os operadores duais “*and*”, “*or*”, “*not*” e as implicações discutidas em seções anteriores. Contudo, a seqüência desses quatro passos é mantida. O processo de raciocínio é *indutivo*, isto é, obedece a um encadeamento progressivo de regras. Os passos aqui esboçados são mínimos para descrição de uma máquina de inferência de regras nebulosas. O detalhamento dos quatro passos, bem como esquemas alternativos para inferências nebulosas, é apresentado na maioria dos textos sobre sistemas nebulosos. As referências [Lee90a, Lee90b, Roc92, Vio93, TS87] são pragmáticas quanto à abordagem de sistemas nebulosos.

4.7.1 1. Passo: “Fuzzificação”

Esse passo descreve um valor (numérico, por exemplo), sendo quantificado sobre as curvas de termos da variável em questão. Como resultante da “*fuzzificação*”, tem-se

um conjunto de termos com seus respectivos graus de pertinência. Esse conjunto é obtido pela interseção do valor com as curvas que definem a variável. A interseção considera um ponto a ser particularizado. Exemplificando: tendo um salário de U\$ 400, uma pergunta típica seria: “qual o grau de satisfação (entre 0 e 1) para variável S ”? Considerando a figura 4.1, o resultado de 400 “fuzzificado” é a seleção de s_1 e s_2 , com $\mu_S(400) = 0.5$ em ambos os termos.

4.7.2 2. Passo: “Matching” ou Casamento

O segundo passo é o “*matching*” (“casamento”) com os termos antecedentes das regras. Essa etapa visa “casar” todos os termos inferidos da “fuzzificação”, com a parte antecedente da base de regras nebulosas. Uma montagem padrão dessas bases com regras do tipo “*se-então*”, tem sido proposta e defendida por Bart Kosko [Kos92, Kos94] pelo esquema FAM (“*Fuzzy Associative Memory*”). Embora não exista nenhum rigor especial quanto a sua implementação computacional, a montagem FAM, tem sido um consenso e acha-se consolidada na prática. Ela visa que todas as regras sejam disparadas paralelamente e parcialmente consideradas sob algum grau de verdade. O disparo das regras é por *casamento de padrões* e *conteúdo* dos termos antecedentes da regra. O esquema FAM visa definir os termos a cada célula de memória do computador. A justificativa do esquema FAM é baseada na viabilidade de implementação física em pastilhas de silício. A proposta é uma base de regras nebulosas em hardware, implementação já existente há algum tempo [Ric88].

Uma vez selecionadas as regras factíveis, surge a necessidade de se calcular o grau de pertinência do(s) termo(s) conseqüente(s). Em Sistemas Baseados em Regras de Produção (SBRP), esse cálculo é chamado de *propagação de crenças*. Normalmente, o operador aplicado em regras nebulosas é o “*mín*” [Kli88], um dos mais usados em esquemas com regras de controle [Lee90a, Lee90b, Miz95]. Esse operador pessimista propaga o menor valor dos antecedentes para o(s) conseqüente(s), embora, na prática, venha apresentando bons resultados.

4.7.3 3. Passo: Mediação

O terceiro passo é a *ponderação* sobre as “ n ” regras disparadas com algum grau de ativação. Essa etapa visa equacionar (contabilizar) o grau de pertinência de cada regra factível. Normalmente, o operador aplicado nesse cálculo é o “*máx*” [Kli88]. No caso, considera-se a área máxima formada pela curva do termo inferido, limitada na altura do $\mu_A(x)$, propagado pelos antecedentes da regra. Caso ocorra mais de um termo inferido, o procedimento se repete igualmente para os demais. Caso ocorram áreas sobrepostas, deverão ser consideradas e computadas. Em [Miz95], encontra-se uma discussão sobre os operadores “*máx*” e “*mín*”. O principal resultado do artigo de M. Mizumoto [Miz95], é que esses dois operadores, em certos casos, podem deduzir uma não-linearidade do sistema, quando no caso real era linear.

4.7.4 4. Passo: “Desfuzzificação”

O quarto passo é a “*desfuzzificação*”. Após a seleção das áreas sobre as curvas dos termos contabilizados no passo anterior, o método consiste em retornar um valor numérico desse cálculo simbólico. O passo resume-se em transformar as ações disparadas em um controle não nebuloso. O passo anterior fornece curvas com uma distribuição aproximada sobre uma resposta. Nesse ponto, o detalhe é encontrar um valor que seja representativo as curvas de conhecimento das regras disparadas. O método mais utilizado é o do *centróide*, ou do centro de gravidade, ou de massa. Esse método é bem aceito como consenso dos pesquisadores [Kos94, Kos92, Sug85]. Contudo, outras proposições podem se fazer valer. O cálculo desse valor é dado por:

$$\bar{y} = \frac{\sum_{i=1}^N \omega_i \cdot y_i}{\sum_{i=1}^N \omega_i}$$

onde N é o número de regras disparadas, ω é o grau de ativação na ação conseqüente y . O valor de ω corresponde à pertinência da ação, portanto $\omega \in [0, 1]$. O método do centróide é razoável, tratando-se de uma quantificação numérica. Caso essa quantificação não corresponda a uma realidade, esse passo tem a sua confiabilidade comprometida.

4.8 Alguns Comentários

Os quatro passos do mecanismo inferencial acima constituem a base para um tipo de modelo de agente descrito no capítulo seguinte. As variações de cada passo são determinadas pelas aplicações em questão. Por exemplo, em [TS87], são apresentadas três variações de métodos inferenciais do raciocínio nebuloso. Num primeiro tipo de solução, as variáveis são do tipo contínuas e discretas. As variáveis contínuas são descritas por funções matemáticas do tipo senoidais. Para que sejam combinadas com as variáveis discretas, um termo de *compatibilidade* é adicionado aos cálculos. Esse termo considera as entradas versus os termos antecedentes da regra. Num segundo tipo de solução, as variáveis têm funções de pertinência do tipo monotônicas. Isto é, as variáveis começam de um valor mínimo com pertinência mínima (ou máxima) e terminam com valor máximo com pertinência máxima (ou mínima). As aplicações que apresentam esse formato de curva em seus termos possuem um número reduzido de regras. Num terceiro tipo de solução, a parte conseqüente das regras possui equações relacionais de suas entradas. Além das regras, a parte conseqüente varia em função das entradas do sistema. Nos três casos acima, o processo de desfuzzificação sofre ligeiras alterações. Contudo, a idéia de contabilizar as várias regras disparadas segundo seu grau de ativação (um peso), permanece ao longo do contexto.

Capítulo 5

Uma Arquitetura para Agentes Autônomos

5.1 Introdução

Neste capítulo é apresentada uma proposta de arquitetura para agentes autônomos (AA's) do tipo móvel. A validação desta arquitetura ocorre pela instanciação de três protótipos, com diferentes funcionalidades. As aplicações são para propósitos diferentes, mas alicerçadas sobre uma mesma estrutura conceitual. Há uma arquitetura de base sobre a qual os protótipos são construídos.

Inicialmente, uma estrutura em três níveis é apresentada à arquitetura de base. Em seguida, os níveis da arquitetura proposta são discutidos sob uma visão da biologia. Na seqüência, é comparada com arquiteturas da área “neuro-simbólica”, mas discutidas na seção por outras razões. Termina-se, enquadrando-a dentro do tema de “*arquiteturas*” para IAD.

Num segundo momento, as aplicações desenvolvidas são descritas, bem como as características físicas do modelo de AA que vai suportar tais protótipos. Ou seja, descrevem-se os elementos físicos que compõem o AA simulado.

Na seqüência, há uma descrição detalhada de cada nível da arquitetura proposta. Em cada um desses níveis, considerações específicas são realizadas cada vez que surgem diferenças entre os protótipos. Finalmente, há um acoplamento dos três níveis detalhados, sob a arquitetura de base originalmente proposta no início do capítulo.

5.2 Arquitetura de Base

A arquitetura aqui proposta é fundamentada em *estruturas* e *esquemas* biológicos [MAA95, Ark93]. As estruturas referem-se à parte física que compõem o ser vivo, enquanto os esquemas referem-se à funcionalidade emergente que ocorre sobre essas estruturas. Uma discussão e exemplos sobre o assunto são encontrados em [MAA95]. A argumentação e a motivação desta arquitetura são alguns princípios da natureza

animal. Sua caracterização visa acompanhar os aspectos *reativo*, *instintivo* e *cognitivo* de um ser biológico. O aspecto reativo se relaciona com sua percepção sensorial imediata. O aspecto instintivo se relaciona com seu comportamento típico ou desejos intrínsecos. O aspecto cognitivo se relaciona com a capacidade do animal desencadear processos mais complexos de seu comportamento típico (por exemplo, a aprendizagem de tarefas particularizadas, como encontrar um objeto utilizando-se do olfato¹), processo esse que é desempenhado a partir de um esquema de memorização do ser vivo. Em resumo, a arquitetura aqui proposta se desenvolve seguindo algumas facetas biológicas.

5.2.1 Os Três Níveis da Arquitetura

Considerando os aspectos reativo, instintivo e cognitivo apresentados por alguns animais, uma arquitetura com três níveis hierárquicos é aqui proposta. A hierarquia é disposta conforme a figura 5.1. As setas correspondem às mensagens (ou sinais) trocadas entre os níveis, e que serão detalhadas oportunamente. Os critérios hierárquicos visam uma boa funcionalidade dessa arquitetura, bem como alguns fundamentos das arquiteturas *orientadas a comportamento* [Mat94] e *puramente reativas* [Bro91c, Bro86]. Nas arquiteturas reativas, essa hierarquia na troca de sinais não é respeitada. Os sinais sensoriais são repassados igualmente a todos os níveis da arquitetura [Ish95].

O nível inferior ou reativo é associado a uma interação direta com o ambiente, tanto na percepção como numa possível ação. O nível intermediário ou instintivo relaciona-se às habilidades “*embutidas*” do comportamento animal. O nível superior ou cognitivo define a funcionalidade dos protótipos de AA’s construídos.

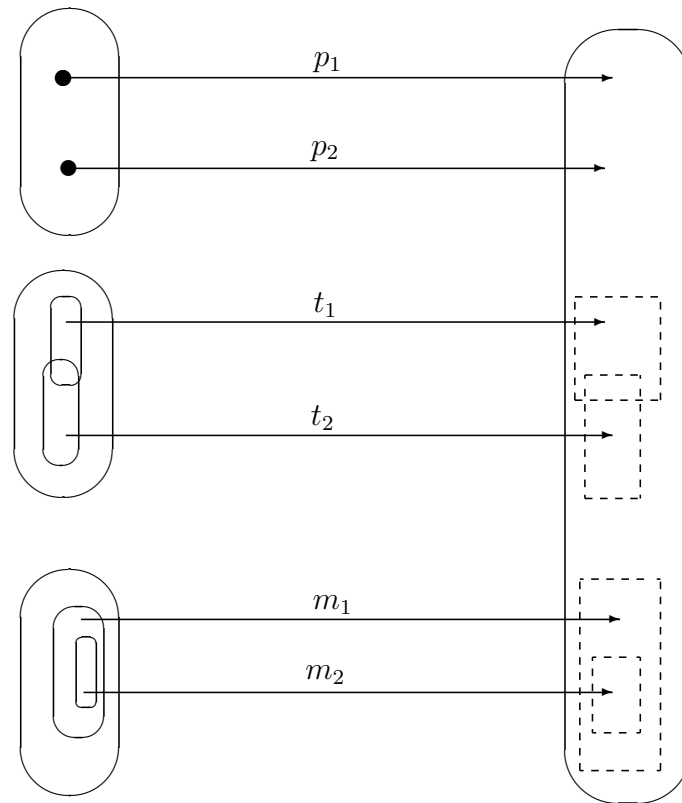
Na subseção seguinte, procura-se qualificar essa arquitetura de três níveis com um enfoque biológico. Finalmente, há um enfoque computacional dessa arquitetura dentro da IAD.

5.2.2 Fundamentação Biológica da Arquitetura

Os seres vivos são formados por vários sistemas. Entre eles encontra-se o sistema nervoso (SN). O SN é formado por dois importantes tipos de tecidos nervosos: o *sensitivo* e o *motor*. O sensitivo é responsável por encaminhar estímulos recebidos pelos *receptores sensoriais* ao cérebro. Esse caminho é encurtado para o caso dos arcos reflexos, onde os receptores sensoriais conduzem os sinais até a medula espinhal. Os estímulos podem ter origem de qualquer campo receptivo, seja pela gustação, visão, olfato, audição, tato, etc. A condução desses “sinais” ao cérebro ocorre por fibras nervosas sensoriais.

O tecido nervoso motor leva o “sinal” do cérebro, que por sua vez controla uma contração ou reação muscular. A resposta é chamada de *função motora* [Guy73], que

¹Característica muito aguçada dos mamíferos em geral. Um estudo sobre essa influência de fluxos químicos sobre insetos e peixes é encontrado em [WK93].



Analogia das três visões (prob, fuzzy, crença)

Figura 5.1: Visão Geral dos Três Níveis dos AA

se manifesta por meio de músculos estendidos e/ou retraídos ou glândulas dilatadas pelo sinal do SN que são chamados de *efetores*. Nem sempre a reação é um movimento. Normalmente, os efetores glandulares provocam uma secreção compensativa. Por exemplo, uma gota de vinagre puro sobre a língua envia um sinal ao cérebro e este envia um sinal ao seu efetor, que são as glândulas salivares para produção de saliva. A rigor, há uma interação complexa envolvendo campos sensoriais, efetores e cérebro. Um dos detalhes está na região formada pelo hipotálamo, tálamo e hipocampo, que são responsáveis pela irrigação de regiões do córtex com endorfinas e, com isso, habilita conexões neuronais que levam a funções mais sofisticadas como: memória (ligada mais especificamente à região límbica), raciocínio, interpretação, etc.

Supondo-se que o sabor do vinagre já era previamente conhecido, há uma capacidade de reprodução de reações químicas que levam a identificar tal gosto. Ou seja, a repetição da produção de endorfinas pelas regiões do hipotálamo e límbica leva a um processo chamado de *memorização*. Eventualmente, as experiências podem ser memorizadas por minutos, dias ou anos, com o objetivo de ter uma reação mais apropriada

a um estímulo já conhecido. Em resumo, são representações sensoriais que levam a uma identificação junto às regiões de memória e córtex, retornando um sinal motor. Se a associação estímulo-ação for reproduzida muitas vezes, pode ser incorporada a um *esquema de memória*. Isto é, o par estímulo-ação passa a ser embutido como uma habilidade inata. A reapresentação de um estímulo conhecido conduz a uma rápida associação no cérebro, retornando a uma ação efetora já executada anteriormente.

A mudança na irrigação a partir da região do hipotálamo² segue o estímulo externo, visando desenvolver processos cognitivos de mais alta ordem nas regiões do córtex. Portanto, a região do hipotálamo é intermediária com relação ao córtex e aos tecidos nervosos. Os sinais que se processam no córtex e os que fluem pelos tecidos nervosos sugerem os níveis cognitivo e reativo, respectivamente. Com isso, há uma aproximação dessas definições com a figura 5.1. O córtex não possui acesso direto ao meio externo, a exemplo do nível cognitivo da figura 5.1. Por outro lado, o córtex possui camadas ou regiões específicas que controlam alguns órgãos e realizam processos cognitivos mais complexos. As regiões mais próximas ao núcleo cerebral (região do hipotálamo) estão ligadas a respostas instantâneas e automáticas diante dos estímulos sensoriais. Já as regiões mais externas do córtex relacionam-se a mecanismos mais conscientes [Guy73, Cri89, Sej89].

O nível intermediário da figura 5.1 visa incluir na arquitetura o aspecto instintivo dos seres vivos. O instinto é definido pela codificação genética do ser vivo desde a célula-ovo ou zigoto (DNA). O instinto pode variar ao longo de um ciclo de vida. Essas variações podem ocorrer por um fator ambiental ou transformações internas (por exemplo, insuficiência glandular) ou, ainda, vir codificadas em sua base genética. Quanto a sua natureza, Freud definiu o instinto como: “*tendências intrínsecas da substância viva à reconstrução de um estado anterior, ou seja, historicamente condicionado de natureza conservadora*” [dP66]. A motivação de Freud era que a compreensão desses instintos levaria a compreensão dos distúrbios de conduta ou comportamento. Contudo, para que o instinto se manifeste há necessidade de uma motivação externa. Ou seja, a base instintiva deve estar diretamente acoplada ao ambiente, visando suas manifestações primárias de sua base biológica. Por essa razão, o nível instintivo da figura 5.1 recebe os sinais diretamente do meio ambiente. Enfim, o instinto é inconsciente e desencadeia-se como um plano internalizado. Ressalta-se, ainda, que o instinto atua com extrema precisão e eficácia. Para que possa se desenvolver fisicamente, há necessidade de se combiná-lo com a atividade sensorial-motora do nível reativo. Em outras palavras, é como se o nível instintivo comandasse um reativo. De acordo com a figura 5.1.

Há uma outra interpretação a ser dada aos três níveis mencionados, considerando o funcionamento cerebral. Parte-se da existência de funções específicas, associadas a regiões do córtex cerebral. Por exemplo, há uma região para o córtex visual no lobo occipital, na região posterior do cérebro, que se preocupa com a recepção e a interpretação das imagens visuais. De acordo com R. Penrose [Pen91], as regiões do

²Há outros órgãos, como o tálamo, o hipocampo, o bulbo e a medula, que são ligados a outras terminações nervosas, como os nervos auditivo, olfativo, etc.

córtex cerebral, relacionadas diretamente com as entradas e saídas do cérebro, são chamadas de *primárias*. Localizam-se próximas aos nervos receptores, ligados diretamente às funções visual, auditiva, olfativa, somato-sensorial e motora. Próximas às primárias, encontram-se as regiões *secundárias*, “*que atuam num nível de abstração mais sutil e complexo*” [Pen91]. As regiões secundárias recebem estímulos sensoriais, processam e enviam um estímulo³ motor a regiões específicas das primárias, não necessariamente à região que deu origem ao estímulo sensorial. O restante do córtex cerebral é considerado como regiões *terciárias*. Nessas regiões realizam-se os processamentos mais abstratos e sofisticados, como a imaginação, o raciocínio lógico, etc. Nas regiões terciárias, as informações são combinadas entre si, processadas, analisadas de forma complexa e enviam estímulos motores às regiões secundárias. Observa-se que há um fluxo sensorial das regiões primárias às secundárias (seguindo-se às terciárias), enviando estímulos motores elaborados a regiões específicas. Observa-se uma geração de ciclos de fluxos entre essas camadas. Tudo ocorre de forma paralela e concorrente, e não necessariamente um estímulo sensorial envia como resposta um outro estímulo motor sobre a região primária que o gerou. A partir das três regiões do córtex cria-se uma hipótese de que o raciocínio humano ocorre também em três níveis. Desse modo, essa é uma segunda analogia com a figura 5.1.

O resumo tem por finalidade enquadrar a figura 5.1 sob um enfoque biológico de um processamento em três níveis, entre os sinais sensoriais de entrada e as respectivas ações motoras de saída. Evidentemente, não é seu objetivo definir um **padrão** para um ser autônomo inteligente a partir destes três níveis assumidos. Procura-se uma coerência com bases biológicas para projetos de AA's. Por exemplo, em termos biológicos é difícil afirmar que o nível instintivo assuma um enfoque intermediário entre processos cognitivos e reativos. Tal nível é uma abstração de processos envolvidos em um comportamento animal.

5.2.3 Considerações sobre a Arquitetura Proposta

Há muito tempo que dentro da IA tem-se tentado realizar uma sinergia com o cruzamento entre os paradigmas simbólico e conexionista [Hil96]. Sob essa visão neuro-simbólica tem surgido propostas de como esses dois paradigmas podem interagir. Em [PA94, Hil96], duas estratégias de interação são colocadas: as *unificadas* e as *híbridas*. As estratégias unificadas visam uma combinação das habilidades neural e simbólica, mas implementadas via redes neuronais numéricas ou artificiais. Os blocos dessa estratégia são redes neuronais numéricas. As estratégias híbridas, combinam redes neuronais e modelos simbólicos, como sistemas especialistas e árvores de decisão. Na estratégia híbrida há *modos de integração* desses componentes, isto é, como são interconectados. Quatro modos básicos são identificados: processamento encadeado (“*chainprocessing*”), subprocessamento (“*subprocessing*”), metaprocessamento (“*metaprocessing*”) e coprocessamento (“*coprocessing*”) [PA94].

Considerando esses modos de interação, a figura 5.1 possui características hierárquicas

³Leia-se “*transformação*” de acordo com a tradução da referência [Pen91].

próximas ao “*metaprocessing*”. Apesar da motivação biológica em torno da arquitetura aqui proposta, essa não faz considerações sobre o uso de redes neurais numéricas. Contudo, são apresentadas de modo oportuno as equivalências em várias instâncias entre essa arquitetura e os sistemas neurais. A rigor, o problema aqui abordado localiza-se em torno da IAD, onde o tema genérico é o de “*arquitecturas distribuídas*” [Gas89]. O tema tem interesses tanto na IAD como na área de projetos de AA’s [Mae91a]. O enfoque dado é em função da granularidade apresentada por seus elementos internos. Os elementos apresentam uma equivalência ao conceito de agente (capítulo 3). Em face à inspiração biológica da arquitetura proposta, os agentes são combinados de forma coerente em torno de topologias neurais, mas internamente possuem um processamento simbólico. Sendo assim, a arquitetura apresentada difere dos conceitos de ser “puramente híbrida” ao aproximar-se de questões das áreas de IAD e AA’s [PA94].

5.3 Apresentação dos Protótipos

Esta seção apresenta a funcionalidade dos protótipos desenvolvidos e instanciados a partir da arquitetura proposta. A ordem de apresentação é a mesma das implementações computacionais (capítulo 6). Esses protótipos foram originalmente discutidos em [dS96].

5.3.1 Brooks

Brooks tem a competência de encontrar uma saída de um labirinto típico. Considerando os labirintos descritos por R. Sun e outros [Mer96, PM96, Pet95], eles são vistos como um único corredor principal e vários becos sem saída nas laterais esquerda e direita. Os tipos de labirintos aqui empregados podem possuir salas, outras saídas, várias formas de paredes (lisas ou irregulares), enfim, labirintos substancialmente mais complexos que os considerados por R. Sun e outros [Mer96, PM96, Pet95]. A restrição é que não podem existir “ambientes” dentro de “ambientes”, em face do sistema de orientação ser exclusivamente local. Por exemplo, uma sala fechada dentro de outra e sem conexões físicas entre elas. Em resumo, esses labirintos devem apresentar um “*caminho físico*” (a continuidade das paredes nas diversas salas) de uma entrada a uma saída.

Ao encontrar-se em uma sala que contenha saída, ela é visualizada como ponto de fim de labirinto ou um alvo a ser atingido. O procedimento adotado para se caminhar no labirinto é aproximar o AA de uma das paredes (esquerda ou direita) e caminhar de modo similar a um ser humano no escuro. Mantendo-se uma das mãos sob a parede, sem trocar de lado, a saída de um labirinto do tipo considerado é sempre encontrada. Portanto, arrastar-se com segurança próximo às paredes é a estratégia adotada para se caminhar no labirinto. O nome “*Brooks*” visa homenagear Rodney Brooks, cujo trabalho apresentou uma “*quebra*” de paradigma em relação à IA simbólica e as *arquitecturas de englobamento* [Bro91c, Bro91b, Bro86].

5.3.2 Steels

Steels visita uma sequência (uma lista ordenada) de locais em um ambiente sem modelamento a priori. Não é feita nenhuma consideração em visitar o local mais próximo de onde o AA estiver. Há uma ordem de locais e essa é seguida. Há uma analogia com os agentes de software (autônomos), que Pattie Maes [Mae95] tem desenvolvido para uso em redes de computadores. Nesse caso, o agente visita nós hospedeiros, em busca de uma determinada informação. Uma analogia é uma viagem por várias cidades sem se conhecer todas elas a priori. A próxima cidade só é conhecida após um tempo de permanência para reabastecimento, etc. O nome desse protótipo é uma homenagem a Luc Steels, pela sua visão multidisciplinar dos AA's, abrange desde AA's com enfoques tradicionalmente simbolista aos puramente biológicos.

5.3.3 Maja

Este protótipo visa contornar algumas restrições de navegação que *Steels* apresenta. *Maja* apresenta um esquema simbólico mais sofisticado, que visa simular o comportamento de um inseto quando ele se encontra em caminhos fechados, como becos, salas sem saídas, etc. Normalmente, para esses casos, surge a necessidade de se realizar um aprendizado, orientado em descobrir um caminho de retorno. Esse problema é de razoável complexidade, que implica em sistemas com memorização e aprendizagem em tempo real [vP94c, vP94b]. Contudo, essa complexidade foi contornada pela modelagem do comportamento de um inseto primário típico. Esse protótipo sofreu alterações na sua descrição original [dS96]. O nome “*Maja*” é homenagem a Maja Matarić, por suas pesquisas em comportamentalismo social, com conceitos da Psicologia, Sociologia, Biologia, aplicados a comunidades de AA's móveis.

5.4 Modelo Físico do Agente Autônomo

O AA aqui estudado é do tipo móvel. Contudo, a metodologia utilizada, bem como a arquitetura de base, podem ser adaptadas ao desenvolvimento de outros tipos de AA's. Por exemplo, braço de robôs, sistemas distribuídos multi-sensoriais, navegações por marcas, etc. Os AA's do tipo móvel apresentam um interesse especial: um largo espectro de aplicações industriais. As várias aplicações na área de robótica credenciam o problema da *locomção* de um robô como um importante problema a ser abordado.

Basicamente, uma *locomção* pode ser vista como um mapeamento entre a informação sensorial versus um estado corrente e uma ação a ser disparada, isto é:

$$l : P \times E \rightarrow A \quad (5.1)$$

onde l é uma função de mapeamento da locomção do agente, P é um conjunto de percepções do agente, E é um conjunto de estados do agente e, finalmente, A é um

conjunto de ações possíveis de ser desencadeados. O tamanho do espaço de situações é o número do mapeamento [RB96], isto é:

$$|A|^{|P||E|} \quad (5.2)$$

Observa-se que a tratabilidade do problema, sob uma visão discreta, torna-se difícil mesmo com poucos estados de P , E e A . Diante dessa dificuldade e outras, como planejamento, sincronização, orientação global, etc, novas abordagens foram surgindo, como os *sistemas reativos* [Bro91c, Bro91b].

Robôs reais apresentam vários tipos de sensores e receptores (ou medidores) de sinais. Os sensores incorporam um papel reativo no contexto dos robôs, enquanto os receptores realizam medições sobre o seu estado corrente. Por exemplo, um dispositivo receptor que indica o nível de corrente na bateria do sistema. Torna-se comum o uso de câmeras de vídeo com objetivo de capturar imagens, decodificá-las e realizar algum processo inferencial sobre elas. Há sensores de alta precisão, utilizando-se raios-laser, que fornecem distâncias exatas entre o robô e um ponto qualquer, sem os problemas de reflexão e refração. Há sensores que são ativados pela percepção de fontes luminosas, marcas significativas, fontes sonoras, etc. Vale ressaltar que sensores e receptores fornecem apenas valores numéricos, sem entendimento algum a respeito de sua semântica [Bro92]. Algumas referências sobre o assunto encontram-se em [FL94, Zel88]. Enfim, há todo um crescimento tecnológico em torno dessa área de sensores, visando uma maior precisão, menor custo, objetivos variados, etc.

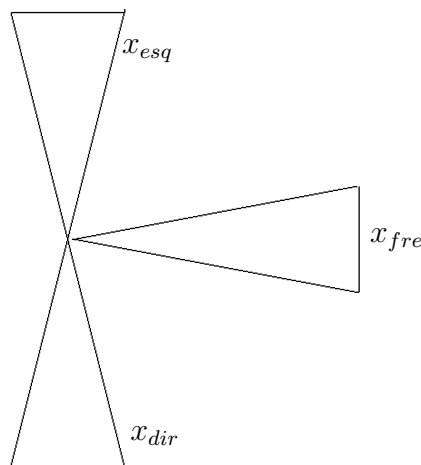


Figura 5.2: Sonares Locais do Agente Autônomo

O protótipo básico do AA aqui tratado possui um número reduzido de sensores, mas com diversas funcionalidades. A disponibilidade dos sensores segue as aplicações. Isto é, certo tipo de sensor pode não ser interessante quando o AA está operando sobre um tipo específico de tarefa, num dado comportamento. A especificação dos sensores, que foram simulados sobre esta arquitetura, é a seguinte:

- 1 - Três sensores do tipo sonares de proximidade. Esses três sonares são do tipo cônicos e as suas localizações no AA são: um na parte frontal, outro na lateral esquerda e outro na lateral direita (ver figura 5.2). Esse conjunto de sensores tem a função de detectar obstáculos nas vizinhanças do AA. Seus valores de entrada são dados por: “ x_{esq} ” para o sonar da esquerda, “ x_{dir} ” para o sonar da direita e “ x_{fre} ” para o sonar da frente. Seu conjunto de valores é dado pelo seguinte intervalo:

$$0 \leq x \leq 1$$

onde o valores próximo a 0 correspondem a uma proximidade do vértice interno do cone e valores próximo a 1 correspondem a um afastamento do vértice interno do cone. Esses sonares têm por objetivo a detecção de objetos, paredes, etc, nas proximidades do móvel. Aumentando-se o ângulo de “visão” ou de abertura do cone, cobre-se uma região de quase 360° em torno do AA. Obviamente que essa varredura na vizinhança do AA poderia ser feita com uma coleção de sensores dispostos em torno de um robô, por exemplo [Bro91b]. Esse é um caso típico em que há muitos sensores do mesmo tipo realizando medidas distribuídas e paralelas, que são integradas numa única medida de *fusão* equivalente. A amplitude (“*Amp*”) é o ângulo de abertura dos sonares e é ajustável por meio de uma interface amigável nos simuladores contruídos. Detalhes da especificação matemática e implementação encontram-se no capítulo seguinte.

- 2 - Um sensor de direção que fornece o desvio angular (ϕ) entre o AA e um alvo qualquer ou uma fonte atratora. O valor de ϕ é o ângulo entre a linha central do sonar dianteiro (a mediana do cone frontal) e um alvo ou marca corrente. A contagem angular é feita em qualquer sentido (horário ou anti-horário), mas com o objetivo de facilitar os cálculos. Os valores são computados em módulo de 180. Portanto, o conjunto de valores ϕ é dado pelo seguinte intervalo:

$$-180^0 \leq \phi \leq 180^0$$

Esse sensor é inspirado na propriedade olfativa dos animais. Os animais tendem a seguir direções fornecidas pelo olfato ⁴, visando fontes de alimentos ou feromônio na época de acasalamentos, etc. Semelhante ao olfato, tem-se a audição, em que os animais se dirigem em direção a uma fonte de estímulo.

- 3 - Um sensor que indica a visibilidade entre o AA e uma marca qualquer, fornecendo uma resposta binária. Uma vez localizado visualmente um alvo, o sensor tem como valor resultante um “*sim*”. Caso contrário, objeto não visível, a resposta é “*não*”. O AA leva em conta esse valor cada vez que o sensor detectar uma

⁴A partir de uma dependência interna dos animais (um est ímulo primário como a fome), que se utilizam de seus sensores naturais para dispararem *comportamentos motivados* [RDBS91], valendo-se também de um estímulo externo.

visão livre entre o AA e uma marca ou uma saída. Formalmente essa visão binária é dada por:

$$\text{visão} = \{ \text{“sim”}, \text{“não”} \}$$

- 4 - Um sensor de distância, d_{xy} , que fornece a distância entre o AA e um alvo qualquer. Uma vez localizado um alvo, o sensor fornece uma medida estimada entre o AA e o objeto. Esse objeto pode estar visível ou não, mas uma medida aproximada pode ser fornecida. Em termos biológicos, a distância entre dois pontos ou uma linha reta nem sempre é levado em conta por um animal na busca de seus objetivos. As escolhas por parte dos animais são geralmente oportunistas, isto é, apenas é considerado a maior facilidade em atingir seus objetivos.

Dos quatro tipos de sensores descritos acima, o sensor de distância (item 4) não foi utilizado em nenhum dos protótipos simulados. Foi implementado, mas não foi útil, pois gerava conflito com relação aos demais sensores. Havia um excesso de dados sensoriais, mas incompatíveis entre si quanto a objetivos comuns. A partir desses dados, define-se um campo sensorial para o AA, como sendo o conjunto de sua percepção (P) da equação 5.1. O campo sensorial ou sensitivo do AA proposto é, portanto, definido por:

$$P \doteq (x_{esq}, x_{dir}, x_{fre}, \phi, \text{visão}) \quad (5.3)$$

Considerando com rigor a equação 5.3, há ausência das coordenadas globais (x, y) , isto é, o lugar em que o AA se encontra em relação ao ambiente. Essa é uma medida embutida ao AA e que numa experimentação física pode ser externa ou de um mapa interno. Para efeitos deste trabalho, essa não é uma medida sensorial e sim de um receptor ou mapa interno.

Um dos principais focos explorados neste trabalho é um AA de poucos sensores com características bem rudimentares. Com poucos e simples elementos sensoriais, construiu-se um sistema de controle para locomoção espacial, de interesse prático e teórico. A locomoção ocorre em ambientes complexos, de modo que o AA é forçado a exibir uma classe de comportamentos robustos. Tudo isso contornando o problema da complexidade computacional, apresentado na equação 5.2.

Finalmente, segue-se a especificação do campo motor desse AA. Dado que o agente possui três sensores de proximidade nas suas três principais direções, o mesmo ocorre com os motores. Idealiza-se um AA com três motores independentes e direcionados da seguinte forma: um à esquerda, um à direita e um à frente. Um acionamento combinado provoca um deslocamento (d), com um desvio angular (θ) de sua trajetória original. A combinação de acionamentos faz o agente descrever trajetórias diversas. Por exemplo, acionando-se os motores esquerdo e o frontal, tem-se como resultante uma curva à esquerda. Eventualmente, se o AA deseja retroceder (andar para trás), basta acionar o motor esquerdo ou direito sucessivas vezes até um giro completo de

180°. Em seguida, aciona-se o motor à frente. Os valores de “ d ” e “ θ ” caracterizam a *ação* do agente propriamente dita como sendo a resposta do AA ao estímulo externo. Formalmente, para esse modelo, define-se a ação “ A ”, pelo seguinte par:

$$A \doteq (\theta, d) \quad (5.4)$$

O fato dos três acionadores (motores) serem independentes leva à questão de como controlá-los e coordená-los, de forma que exista uma ação final combinada e coerente em relação a um contexto. A independência evidencia o caráter desta arquitetura ser distribuída em seus processos.

Retomando-se a equação 5.1, falta definir o conjunto de estados (E) em que o AA pode se encontrar. A arquitetura aqui apresentada é do tipo multiagente, onde internamente vários agentes se encontram competindo e cooperando na ativação de uma única ação. Há uma seleção competitiva interna do AA, em que o agente mais habilitado tem o direito a disparar uma ação. Contudo, os agentes internos possuem um espectro muito amplo quanto a seus níveis internos e intensidades de disparo, os quais apresentam valores contínuos. Observa-se uma dificuldade (para não dizer, uma inviabilidade) em formalizar a definição do conjunto de estados do AA a partir dos seus agentes internos.

A alternativa é pensar em termos de valores de uma “*memória corrente*” do AA, buscando uma caracterização em vários momentos sucessivos, mas sem considerar o seu campo sensorial. Para isso, necessita-se encontrar uma seqüência típica da locomoção (l)⁵ do AA, que conduza à constatação da permanência em um estado. O ponto central é que a seqüência não deve ser alterada a partir de uma única leitura sensorial, mas sim a partir de várias entradas atípicas. Isto é, até a ocorrência de uma “*ruptura*” no padrão de entradas. Durante esse intervalo de tempo, o AA assume um comportamento típico, que é alterado em seguida por um conjunto de entradas atípicas. Portanto, esses períodos de “*longo prazo*” definem os estados comportamentais característicos assumidos pelo AA. Em resumo, associa-se um estado do AA ao seu comportamento por períodos a “*longo prazo*”, cuja transição ocorre em face de uma ou mais entradas atípicas.

Observa-se que o conjunto de estados (E) e das entradas (P) derivam uma descrição análoga a um autômato de estados finitos. Desse modo obtém-se:

$$E = \{e_1, e_2, \dots, e_n\} \quad (\text{n-estados})$$

e

$$P = \{p_1, p_2, \dots, p_m\} \quad (\text{m-entradas})$$

Uma transição entre estados δ é definida por:

$$\delta(e_i, p_j) = e_k$$

⁵Da equação 5.1.

Por sua vez, a funcionalidade do AA encontra-se diretamente ligada ao conceito do comportamento a “*longo prazo*” acima descrito, de modo que cada protótipo implementado possua um *autômato comportamental* em particular. O *autômato comportamental* é apresentado no nível cognitivo de cada protótipo.

Em resumo, o conjunto P apresenta um espectro contínuo de seus valores numa faixa numérica potencialmente infinita (ver equação 5.3). O fato torna inviável a tratabilidade do problema quanto ao número de transições possíveis. Contudo, ressalta-se que apenas alguns eventos externos significativos são responsáveis por alterar o comportamento do AA. A idéia é que o AA acompanhe a sua entrada sensorial. Desse modo, o problema é contornado considerando-se apenas o que de fato faz o AA mudar de comportamento.

Considerações físicas sobre o AA como velocidade, peso, aceleração etc, não são itens relacionados diretamente a este trabalho. Contudo, estudo sobre trajetórias versus as questões da dinâmica newtoniana podem ser encontrados em [Gir88, Tou88].

Outra importante questão refere-se ao ruído a que as implementações físicas estão sujeitas. O ruído é um sinal involuntário ao sistema, de características não-lineares, que de algum modo altera as leituras sensoriais de um AA físico. O ruído, eventualmente, pode exibir um sinal mais forte que o valor real desejado. Contudo, este trabalho é desenvolvido em ambiente simulado, portanto isento de certas questões práticas dos AA's.

5.5 Nível Reativo

O primeiro e o mais baixo nível desta arquitetura é o *sensorial* ou *reativo*. Os três protótipos, *Brooks*, *Steels* e *Maja*, usam esse nível sem nenhuma mudança conceitual ou de implementação. O nível reativo tem uma motivação biológica sobre os canais receptores dos seres vivos. O nível apresenta uma rede de agentes, que combinados entre si, acarretam numa *funcionalidade emergente* [Ste94b]. A inspiração dessa rede é biológica, e a principal característica é o fato de ser distribuída. A característica de sistemas distribuídos é cada agente ter acesso direto aos dados sensoriais, mas a partir de um conjunto independente de sensores [Car95]. Os agentes aqui modelados incorporam individualmente uma máquina nebulosa para raciocínios aproximados.

Inicialmente, realiza-se o estudo de um circuito neuronal simples. A partir desse circuito simples, evoluiu-se conceitualmente para uma rede ou um circuito de agentes. Em seguida, um agente nebuloso de interesse é formalizado, visando a ampliação do número de conexões em seus circuitos básicos. Os circuitos são estendidos, até que apresentam um conjunto de circuitos que atendam aos requisitos sensoriais e ativos do AA proposto.

5.5.1 Conexões Neurais

O neurônio é uma célula do sistema nervoso. Seu formato e função variam de acordo com sua localização no cérebro. Os neurônios de interesse estão localizados junto

ao córtex cerebral. Contudo, sua operacionalidade é omitida neste texto em face da vasta literatura existente sobre modelos computacionais inspirados em neurônios biológicos [Car92, Roc92, Lip87]. O interesse reside sobre como ocorrem as conexões neuronais, com objetivo de fundamentar e estabelecer uma arquitetura neuronal, que reflita o nível reativo da arquitetura aqui proposta.



Figura 5.3: Circuito Neuronal - Unidirecional

Inicia-se com uma análise de conexões e topologias neuronais. Por exemplo, o circuito neuronal da figura 5.3. Esse circuito é uma simplificação da modelagem neuronal de um caramujo oceânico (*Aplysia*), reapresentado por J. Vaario e outros em [VO92]. Em [VO92], o modelo da rede neuronal equivalente ao caramujo possui três neurônios sensoriais na entrada, que resumem os sinais de sua cauda, da cobertura e dos sífões (antenas frontais). Há um segundo neurônio intermediário modulador e um neurônio motor na saída. Os três neurônios da entrada possuem conexões diretas com os neurônios intermediários (central ou facilitador) e motor. No neurônio motor observa-se um acoplamento direto da entrada com a saída. O neurônio central descrito [VO92] possui conexões recorrentes, detalhe inexistente no circuito da figura 5.3.

Os neurônios na figura 5.3 são representados pelos círculos e possuem a sua saída axonal ligada à entrada dendrítica do neurônio seguinte. O neurônio #1 tem uma saída axonal com o valor “a”, ligado aos dendritos do neurônio #2, o qual possui um sinal no axônio de saída igual a “r”, que é a entrada nos dendritos do neurônio #3. Finalmente, o neurônio #3 apresenta o valor “m” como saída axonal do circuito neuronal. O axônio com o valor “x” é uma fibra (ou um conglomerado de fibras) de entrada, cujo estímulo sensorial pode ter origem em algum *campo receptivo* [Guy73]. Assim como os outros sinais, ele pode ter origem em uma somatória espacial e temporal de outros neurônios secundários, que eventualmente estejam ligados a essa conexão por meio de seus dendritos. Em analogia com o circuito de [VO92], o valor axonal “m” do neurônio #3 é uma saída que representa um *campo motriz* do circuito da figura 5.3. Oportunamente, sobre esse circuito é feita uma interpretação para os símbolos “x”, “a”, “r” e “m”.

Em redes neuronais numéricas, os valores “a” e “r” correspondem aos pesos finais obtidos de um processo de treinamento. O conjunto de pares de “x” e “m” correspondem ao conjunto de treinamento a que a rede está sujeita. Em redes neuronais simbólicas [Roc92], esses valores se equiparam a símbolos a serem trocados por nós neuronais. Nas redes neuronais simbólicas, os neurônios são modelados de modo a terem uma capacidade de processamento equivalente a uma Máquina de Turing

[Arb69] (ou seja, um processador genérico). A ênfase que A.F.Rocha [Roc92] faz com relação aos neurônios simbólicos é que eles possam sumarizar as atividades neuronais de uma região do córtex. Uma das propostas é flexibilizar computacionalmente as funções de neurônios e regiões, comparando-se aos modelos artificiais existentes nas RNA's. Apesar desses neurônios serem excessivamente complexos quando comparados a um neurônio típico, A.F. Rocha [Roc92] é rigoroso nos critérios biológicos quanto ao *comportamento* de um neurônio natural. Sob tais requisitos, encontram-se as definições de *transmissores* (**t**) (neurotransmissores liberados pelo axônio, relacionam-se a condições pré-sinápticas), *receptores* (**r**) (neurotransmissores presentes no corpo celular, relacionam-se a condições pós-sinápticas), *moduladores* ou *controladores* (**c**) (neurotransmissores facilitadores da sinapse, que avaliam as afinidades entre **t** e **r**) na sinapse dos neurônios naturais, que desencadeiam uma *ação* como resposta neuronal. Ou seja, o neurônio possui uma ação a ser desencadeada segundo as quantidades críticas de **t**, **r** e **c**, presentes no neurônio.

De maneira análoga à definição dos neurônios simbólicos de A.F. Rocha [Roc92, MA94], denominam-se os nós da figura 5.3 como *agentes*. Os agentes provocam uma transformação de variáveis e também *agem* sobre os valores transmitidos ao longo desse circuito. Seguindo um comportamento biológico qualquer, os agentes passam a ser equivalentes aos neurônios simbólicos [Roc92, MA94]. Algumas idéias sobre teorias e definições sobre agentes são encontrados no capítulo 3. Contudo, antes de se analisar circuitos mais complexos de agentes que o da figura 5.3, é oportuno especificar os agentes a serem utilizados no nível reativo. Há necessidade de se conhecer a sua granularidade e funcionalidade.

5.5.2 Agentes Nebulosos

A linha de pesquisa que aborda redes neuronais artificiais (RNA's) associada à lógica nebulosa é chamada de “*neuro-nebulosa*” [SV93, Wak93, Tsc94, GR95]. Por sua vez, as RNA's possuem uma estreita relação com a biologia. Apesar da inspiração biológica na arquitetura aqui apresentada, sua abordagem encontra-se inserida na área da IAD. Isso se deve ao fato da exclusão dos *algoritmos de treinamento*, típicos das RNA's, na arquitetura em foco. Desse modo, a questão restringe-se a análises de *topologias* e tipos de *agente* (inicialmente denominado por neurônio). O encaminhamento em torno da IAD faz com que exista a necessidade de se definir um agente nesse contexto. Em face às propriedades apresentadas pela lógica nebulosa, o agente é definido segundo um sistema de produção do tipo nebuloso. Considerando os questionamentos em torno de uma descrição de agente⁶, o agente nebuloso definido para o nível reativo é formalizado por:

$$Agente_k = (X_k^m, \Psi_k, Y_k^n) \quad (5.5)$$

onde os termos da tripla são os seguintes:

⁶Ver capítulo de 3.

X_i^m : é o conjunto das entradas (alfabeto, código, etc) transmitidas ao *Agente_k*. Os seus graus de ativação μ_i (limiares mínimos para disparo) são obtidos a partir da fuzzificação dos m-ésimos termos da variável X_k . O uso da variável “ X ” como mensagem de entrada em um agente visa manter coêrencia com a parte antecedente das regras, da notação empregada no capítulo de lógica nebulosa;

Ψ_k : é o conhecimento interno do *Agente_k*. A representação de conhecimento é embutida no agente e dada pelo seguinte mapeamento:

$$\Psi_k = \{X^m \rightarrow Y^n\}$$

Vale ressaltar que a lógica nebulosa permite modelar um conhecimento, sem fazer considerações sobre a relação entre o número de termos na parte antecedente (m), com o número de conseqüentes (n). O limite superior do número de regras é um produto cartesiano de $(X^m \times Y^n)$;

Y_j^n : é o conjunto de saídas (alfabeto, código, etc) transmitidas pelo *Agente_k*. Os graus de intensidade das saídas são dados por μ_j . A partir do conjunto das saídas busca-se um único valor mediador, que é obtido por algum método de desfuzzificação. Por exemplo o centro de massas [Lee90b].

Resumindo, a definição do conhecimento de um *Agente_k*, segundo a formalização dos sistemas de regras de produção (sistemas de Post), é formado por um conjunto de regras do tipo:

$$\Psi_k = \{X_k^m \longrightarrow Y_k^n\}$$

Agentes com essas descrições (ver equação 5.5) apresentam uma particularidade quando utilizados em sistemas de controle [Lee90a, Lee90b]: a *suavidade* de suas funções compensatórias. Outro detalhe é a possibilidade de se mapear qualquer função matemática e/ou relação algébrica. São características que nortearam a escolha de agentes nebulosos no nível reativo da arquitetura.

5.5.3 A Semântica dos Agentes

Uma vez definido os agentes, retoma-se a análise do circuito da figura 5.3 para uma interpretação de seus símbolos. A figura 5.3 pode agora ser vista como um circuito de três agentes em camadas. A analogia interpretativa para um agente destinado à captação de sinais periféricos (#1), outro para um processamento central numa região do córtex (#2) e, por último, um agente muscular desempenhando uma ação motora (#3). A interpretação é próxima ao trabalho de J. Vaario e outros em [VO92], que reforçam a modelagem computacional de funções biológicas.

Uma outra interpretação é dada em termos de elementos físicos, como distância, energia, etc. Considerando-se os rótulos “x”, “a”, “r” e “m” como variáveis do tipo:

“distância” (x), “andar” (a), “recursos” (r) e “motora” (m), respectivamente. O circuito de agentes traduz uma informação semântica do tipo: “há uma certa distância, que precisa ser percorrida (andando), necessitam-se de alguns recursos, e que para tal efeito, resultam numa ação motriz”. Observa-se uma montagem condicional de regras de produção com encadeamento à frente (“forward-chaining”). O agente #1 possui o valor “ x ”, que representa uma distância a ser percorrida, que é condicional à ação de andar como conseqüente. O agente #2 tem a condicional de andar “ a ”, mas implica em requisitar recursos. Disponibilizando-se recursos “ r ” na entrada do agente #3, uma ação motriz é desencadeada pelo circuito. Sob uma outra ótica, o circuito de agentes traduz uma idéia de deslocamento entre dois pontos. Contudo, o detalhe a ser observado é o desmembramento segundo alguns elementos básicos (partes menores). A existência de uma distância, implicando no fato de andar, que leva a gastar recursos para se gerar uma ação motriz.

Uma outra interpretação é por meio da lógica nebulosa. Exemplificando: se a distância for grande é preciso andar muito. Para se andar muito, gastam-se alguns recursos. Gastando-se tais recursos, gera-se uma ação motora. O circuito conecta um estímulo sensorial de entrada (noção de distância “ x ”), até uma ação motora de saída (ação motriz “ m ”). Há uma proximidade conceitual de causa-efeito produzida pelo sistema nervoso. A distância é um estímulo detectado via um sensor, enquanto que a força motora inferida, provoca um dado deslocamento. As noções de distância, como próximo, médio, longe, etc, representam uma *suavidade* em descrever esses elementos. Sendo assim, a lógica nebulosa se qualifica a realizar tais descrições.

Da figura 5.3, observa-se a possibilidade de um mapeamento direto $X \rightarrow M$, ou seja, no caso do par (X, M) , uma relação direta entre seus termos é factível de ser estabelecida. Contudo, essa “quebra” em partes menores e relevantes deu novas características ao problema:

- Possibilitou um paralelismo a nível de processos. Cada nó poderia estar sendo resolvido por um processador diferente;
- Aumentou a confiabilidade da ação, a qual tornou-se mais compreensível e detalhada. Por exemplo, uma ação motora ocorre pela disponibilidade de recursos. O elemento recurso pode estar associado há um outro esquema externo ao circuito, provocando, assim, critérios de dependência à ação;
- Diminuiu a possibilidade da existência de tarefas inúteis e redundantes. A compreensão das partes menores tornou o estudo mais transparente a nível de tarefas;
- Aumentou a possibilidade de compartilhar tarefas com outros circuitos. Isso significa que um agente pode tomar parte de dois ou mais circuitos distintos (assunto tratado nas seções seguintes), formando subcircuitos.

Em resumo, houve um ganho semântico significativo na compreensão das partes menores. O desmembramento remonta à idéia da IA clássica, da divisão “top-down”

em subsistemas. Os itens acima dão um novo encaminhamento ao problema para dentro da IAD, mais especificamente na “Solução Distribuída de Problemas” [Bit96]. Apesar dos mapeamentos explicitados serem simples, isso pode não ocorrer em sistemas intrinsecamente distribuídos. Os sistemas que exibem uma distribuição espacial, certamente exigiriam uma solução também distribuída. As observações acima tornam-se válidas e reforçadas com os circuitos subseqüentes.

5.5.4 Outro Circuito

O circuito da figura 5.3 não apresenta dificuldades de interpretação, afinal, o fluxo é do tipo “*forward*” (análogo às redes neurais biológicas), e o mapeamento entre os vários pares de entrada e saída de cada agente são praticamente realizáveis. Contudo, as redes biológicas não apresentam tal simplicidade de análise, uma vez que cada neurônio se liga a centenas de outros neurônios [Guy73].

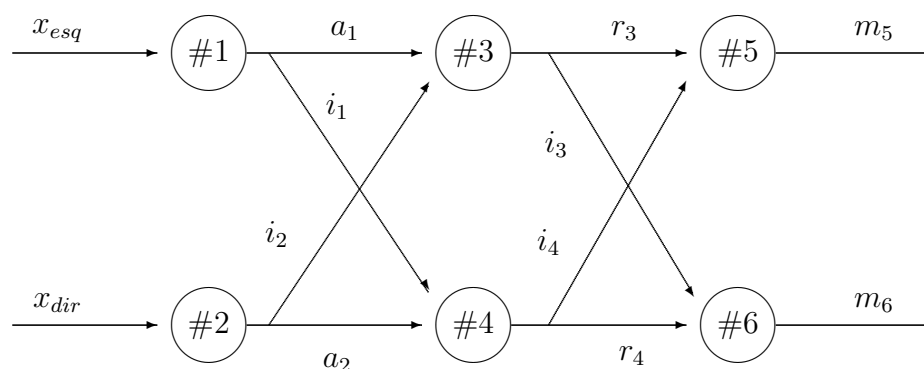


Figura 5.4: Circuito de Agentes - Bidirecional

Em direção a um aumento na complexidade de circuitos neurais, pode-se interligar dois circuitos da figura 5.3, resultando no circuito da figura 5.4. Se os campos receptores x_{esc} e x_{dir} estiverem opostamente orientados, assim como os agentes motores, há uma maior evidência no aspecto da distributividade do circuito 5.4. Nesse caso, os agentes perceptuais #1 e #2 recebem sinais sensoriais dos lados esquerdo e direito respectivamente, que por sua vez, atuam sob os agentes motores #5 e #6 nessas direções. Os dois circuitos estão duplamente conectados entre si por axônios rotulados com “*i*” em seu sinal (ver figura 5.4). Os axônios que conectam dois circuitos enviam sinais do tipo “*secundários*” em relação aos sinais “*primários*”, mostrados no circuito da figura 5.3. O sinal secundário “*i*” é do tipo *inibidor*, o que difere das saídas primárias de cada agente “*a*”, “*r*” e “*m*”. Sob a ótica da matemática, inexistem funções que apresentem uma entrada com duas ou mais saídas. Contudo, tal fato é largamente encontrado em neurônios naturais. Geralmente, essas ligações ocorrem

por intermédio de neurônios auxiliares que funcionam como *mediadores* (inibidores ou excitadores) [Guy73] entre circuitos, permitindo um estímulo ou uma inibição de outro sinal na entrada. Um dos mapeamentos matemáticos possíveis é representar uma função via um conjunto de regras de produção do tipo nebuloso, permitindo que uma entrada produza efeito em paralelo de uma ou mais saídas.

Quanto aos valores dos sinais inibidores, atribui-se valores análogos às saídas primárias de seus agentes. Por exemplo, se uma saída “ a_1 ” em #1 é alta, o valor de “ i_1 ” na saída em direção ao outro ramo também é alta. Se há um sinal elevado para “ i_1 ”, então o outro ramo tende a diminuir a sua atividade no agente #4, pois o sinal “ i_1 ” é alto ou forte o suficiente para inibir a atividade desse agente. Em #4, os sinais primário “ r_4 ” e secundário “ i_4 ”, têm valores baixos em face de “ i_1 ” que inibiu tal agente. Se “ i_4 ” é baixo, então o mesmo não surte efeito na presença de “ r_3 ”. Como saída, tem-se um valor de “ m_5 ” maior que “ m_6 ”. Em face a essa resposta motora, o deslocamento dar-se-á de modo predominante à esquerda. Em resumo, a simetria estrutural do circuito leva a um equilíbrio natural. Se ocorrer um sinal predominantemente mais forte em uma das entradas, tal ramo irá minimizar a atividade do outro. Conseqüentemente, as saídas geram valores desiguais em cada ramo.

Contudo, observa-se que se os agentes #1, #3, e #5 forem iguais aos #2, #4, e #6, respectivamente; com as entradas “ x_{esq} ”⁷ e “ x_{dir} ”⁸ iguais, a ação diferencial desse circuito é nula. O circuito possui uma topologia simétrica, portanto, tal fato torna-o inerentemente conflitante. Para se resolver conflitos em circuitos simétricos, alternativas de interesse são sugeridas:

- Aplicar esquemas de prioridade a um dos ramos do circuito. Isto é, privilegiando-se as atividades dos agentes em um dos ramos perante os demais. Tal estratégia é adotada em parte nesta tese;
- Realizar uma consulta externa (um agente extra de um outro nível) sobre a preferência de atuação de “ m_5 ” ou “ m_6 ”, de acordo com objetivo do sistema;
- Realizar escolhas aleatórias entre os ramos competidores. Essa estratégia é de baixo custo, mas pode divergir dos objetivos globais do sistema.

Essas estratégias visam primordialmente “quebrar” o equilíbrio de circuitos simétricos em geral. Sob a visão da IAD, há algumas resoluções que poderiam ser encaminhadas [Kle91, AJ95]. Um circuito bem mais simples que os da figura 5.4 foi utilizado em [Tor96b, Tor96a] para se misturar sinais primários, objetivando uma recuperação de suas fontes originais. Nesses trabalhos, realiza-se uma análise matemática de duas equações diferenciais não-lineares na obtenção dos sinais de saída. Em [Tor96b, Tor96a], não há qualquer preocupação com questões de conflito, pois os valores de saída são independentes e podem coincidir.

⁷Valor do sinal captado pelo sensor da esquerda.

⁸Idem à direita.

Contudo, tratando-se de um sistema de deslocamento com base no circuito compensador de duas entradas e duas saídas, tal questão é pertinente. Enfim, o importante é ressaltar a inerência de conflitos existentes em circuitos com topologias dessa natureza. Outro detalhe implícito no circuito da figura 5.4 é o controle atuador inerente. Por exemplo, se for apresentado uma sequência contínua de valores altos no ramo esquerdo e valores baixos no ramo direito, a ação resultante é um movimento predominante à esquerda. Sendo assim, ao se inverter gradualmente os valores, a ação irá diminuir lentamente, passando pelo estado de parado e retomando com um movimento à direita.

5.5.5 Estendendo os Circuitos

Nas figuras 5.3 e 5.4, tem-se um e dois receptores sensoriais, respectivamente. A ampliação do problema para três campos sensoriais, à esquerda, à direita e um à frente, resulta em circuitos típicos aos das figuras 5.5 e 5.6. Antes da discussão semântica destes circuitos, resalta-se a analogia direta que eles apresentam em seus campos receptores⁹, com os sinais capturados pelos cones sensoriais no modelo do AA estudado (ver figura 5.2). Os cones fornecem noções de distâncias em torno do agente. A modelagem é bastante plausível, face à semelhança da parte sensorial e motora dos circuitos com o modelo de AA definido anteriormente. Diferentemente de sensores que captam imagens, os sensores aqui considerados fornecem valores reais a uma proximidade ou não do AA em relação a um obstáculo qualquer.

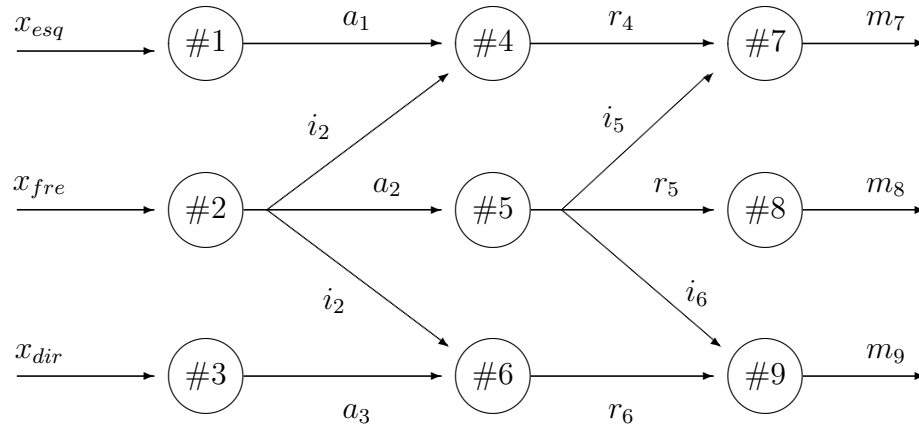


Figura 5.5: Circuito de Agentes (Controle à esquerda, à direita e à frente)

O circuito da figura 5.5, tipo “*espinha de peixe*”, pode apresentar algumas interpretações semânticas quanto as suas entradas. Por exemplo, considerando o ramo

⁹Isto é, os agentes #1, #2, e #3, das figuras 5.5 e 5.6.

central com a entrada “ x_{fre} ”, assume-se algo do tipo: “caso haja um forte sinal sensorial à frente, esse ramo procura maximizar uma ação motora final nessa direção e inibir os ramos vizinhos da esquerda e da direita”. A ação resultante é uma quantidade de movimento à frente. Caso o sinal de entrada em “ x_{fre} ” for fraco, ele pode comunicar aos ramos vizinhos que manifestem suas ações motoras à esquerda ou à direita. No caso, há uma baixa atividade por parte dos agentes no ramo central, permitindo uma predominância dos ramos esquerdo ou direito. Observa-se uma simetria desse circuito, onde as conexões entre os ramos são formadas pelos arcos com rótulos “i”. A orientação dos arcos secundários são propositalmente dirigidos a dar uma funcionalidade coerente, utilizando-se de uma combinação explícita entre os agentes. Obviamente, mais conexões poderiam estar presentes entre os ramos. Contudo, uma funcionalidade coerente não seria diretamente atingida tão facilmente.

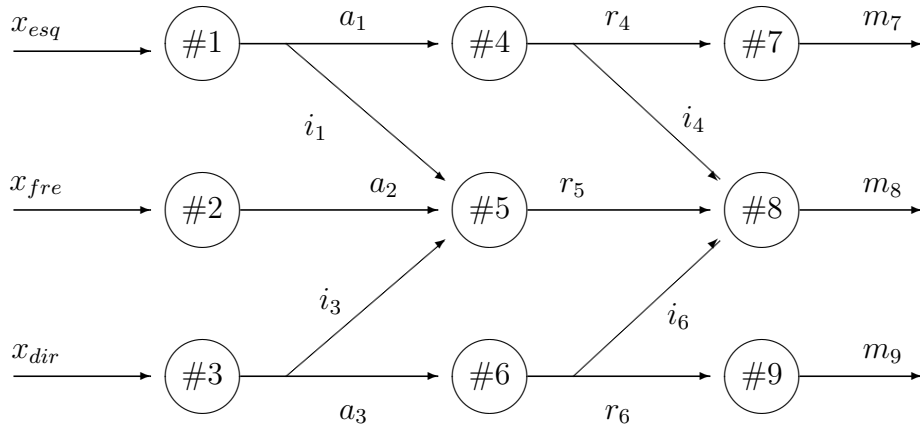


Figura 5.6: Circuito de Agentes (Controle à esquerda, à direita e à frente)

O circuito da figura 5.6, tipo “*seta siga em frente*”, tem no ramo central a recepção dos sinais dos ramos da vizinhança à esquerda e à direita. Objetivando uma ação natural com um movimento à frente, há necessidade que ramo ou subcircuito central (agentes #2, #5, e #8) apresentem alguma prioridade. Considerando um movimento à frente, uma possível semântica é dada por: “caso exista um sinal forte de x_{fre} , esse ramo ignora os sinais oriundos de sua vizinhança”. Caso o sinal seja fraco em “ x_{fre} ”, os ramos da esquerda e direita competem por uma ação motora predominante. Por outro lado, uma prioridade ao subcircuito esquerdo faz com que conexões se direcionem ao subcircuito direito, com o objetivo de inibir ou impor a sua condição de favorável numa ação motora. Em síntese, mudanças nas conexões entre os ramos geram variações de circuitos, cuja funcionalidade resultante pode caracterizá-los como equivalentes entre si (ver circuitos 5.6 e 5.5).

Os circuitos das figuras 5.5 e 5.6 podem ser recombinaados resultando em variações topológicas. Algumas variações possíveis são mostradas nas figuras 5.7 e 5.8. Todos

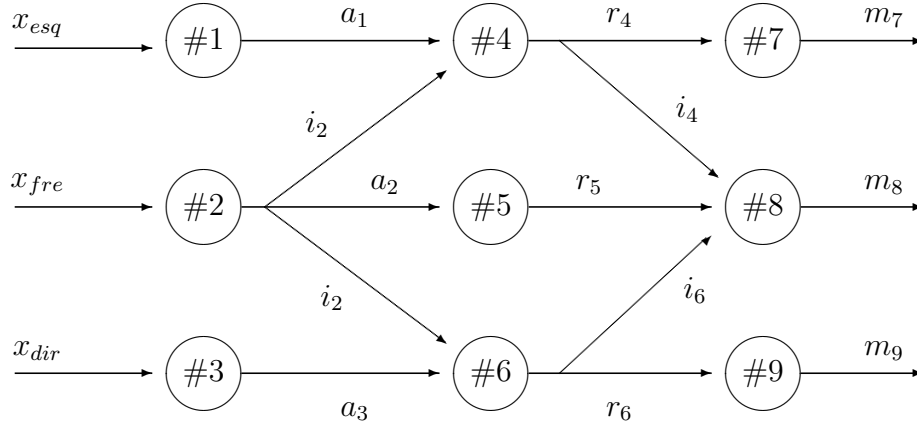


Figura 5.7: Circuito de Agentes (Controle à esquerda, à direita e à frente)

os circuitos, figuras 5.5, 5.6, 5.7 e 5.8, são topologicamente diferentes, mas *possuem a mesma funcionalidade*. Considerando-se a dominância de um movimento à frente aos quatro circuitos no ramo central e combinando-se adequadamente as bases de regras de cada agente, os circuitos apresentam uma mesma função comportamental. Contudo, a base nebulosa de cada agente nos circuitos é variável, não só pelo número de entradas e saídas, mas também pela sua semântica interna desejada. Outro detalhe refere-se a contínua adaptação desses circuitos quando submetidos a variações em suas entradas. A capacidade de adaptação é intrínseca a essas topologias, porém com inerentes problemas de conflitos. Portanto, há necessidade de se priorizar um dos ramos¹⁰, visando uma ação combinada de suas forças motoras independentes.

Retomando-se a especificação de modelo do AA, detecta-se uma analogia do aspecto sensorial e motor aos circuitos das figuras 5.5 a 5.8. Há três receptores independentes, equivalentes aos cones sensoriais do modelo, que controlam de forma compensativa e adaptativa, três motores também independentes. As ações motoras combinadas à esquerda, à frente, e à direita, provocam uma ação tipicamente *emergente* [Ste94b]. O AA modelado sob um desses circuitos (ou arquiteturas) de controle gera resultantes físicas, que leva a um deslocamento sob uma dada direção. Enfim, uma ação motora que combine as saídas dos agentes #7, #8 e #9 leva a um deslocamento “ d ” numa dada direção θ . As formalizações de “ d ” e θ são definidas pela expressão 5.4.

Quanto à equivalência dessas arquiteturas de controle, observa-se que a complexidade computacional é da mesma ordem. Há um mesmo número de agentes, arcos e tipos de conexões em todas as figuras. Em todos os circuitos apresentados, o problema do conflito permanece, sendo inerente aos circuitos. Contudo, optou-se por dar

¹⁰Reforçando, para o caso das figuras 5.5 a 5.8, a prioridade foi dada ao subcircuito central, desejando-se um movimento à frente.

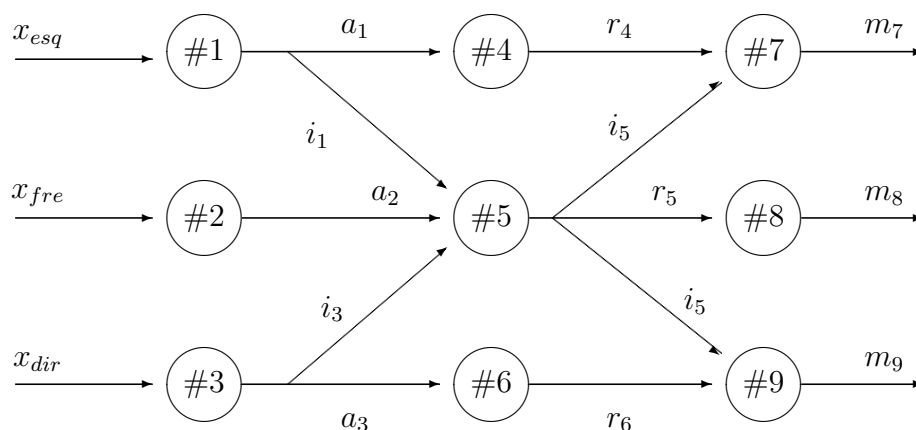


Figura 5.8: Circuito de Agentes (Controle à esquerda, à direita e à frente)

uma prioridade ao ramo central, que regula o movimento à frente. Um esquema de prioridade foi estabelecido sob todas as bases de regra de cada agente, para que os agentes do ramo central tenham preferência em seus sinais. Efetuando-se tal critério de preferência, há um desequilíbrio no circuito, gerando um sinal mais forte em uma das saídas (“ m_7 ”, “ m_8 ”, ou “ m_9 ”).

O circuito da figura 5.7 foi apresentado em [MA94] como um sistema de atuação inteligente em controle de processos. Contudo, sem resultados de seu uso. Em função do circuito exibir três saídas no agente # 2 e três entradas no agente #8, o circuito reúne um maior número de *tipos de conexões*. Em razão da diversidade de conexões em um mesmo circuito, a figura 5.7 foi escolhida para estudo do nível reativo da arquitetura aqui proposta (ver figura 5.1). Os detalhes de implementação do circuito 5.7 encontram-se no capítulo seguinte e apêndice.

5.5.6 Algumas Considerações

Os circuitos discutidos anteriormente tiveram suas complexidades aumentadas gradativamente. Inicialmente, descreve-se um simples circuito unidirecional, seguindo por um bidirecional e, finalmente, um circuito com saída nas três direções (à esquerda, à direita e à frente). Dessa evolução, surge a pergunta: “há um circuito que receba um sinal do lado oposto ao frontal?”, isto é, na região posterior do móvel. A resposta é sim. Tal circuito já foi pensado como uma combinação de um circuito unidirecional simples e um dos apresentados nas figuras 5.5, 5.6, 5.7 e 5.8. Contudo, constatou-se que a controlabilidade de tal arquitetura é complicada e até contraditória. Alguns agentes reativos ou nebulosos devem dispor de “*bases mutantes*”. Ora um conhecimento seria válido para um contexto, ora, ele deveria ser suprimido para dar vez a um novo tipo de conhecimento. Em resumo, os agentes variam circunstancialmente

suas bases de conhecimento. Provavelmente, uma análise sobre circuitos, que geraria uma outra tese.

Outro ponto é a questão do conflito, um dos problemas mais antigos dentro da IA. Novamente, encontra-se diante de um limite natural: os conflitos existentes em circuitos simétricos. Necessita-se de uma abordagem que tire alguma vantagem desse impasse. Optou-se em criar mais um nível, acima do reativo, com o objetivo de prover uma auto-suficiência ao AA móvel. Esse nível é o *instintivo* ou intermediário da arquitetura em questão. Ver figura 5.1.

Em resumo, nessa seção foi apresentada uma arquitetura de controle para o nível reativo, na qual a figura 5.7 foi escolhida. Os agentes nesse nível foram formalizados por meio da expressão 5.5. Com isso, definiu-se a granularidade dos agentes internos ao nível reativo da arquitetura do AA aqui apresentada.

5.6 Nível Instintivo

O objetivo desse nível intermediário é orientar o AA a executar ações mais eficazes, de acordo com a sua funcionalidade pré-definida (ver figura 5.1). Os circuitos das figuras 5.5 a 5.8 controlam de forma aceitável uma navegação sem comprometimentos com tempo, recursos e desempenho. A proposta do nível instintivo é atuar sobre o reativo, tornando-o mais produtor em suas ações. O melhoramento das ações ocorre via uma seleção de *esquemas*¹¹ a serem disparados sobre o nível reativo. Basicamente, o esquema busca um melhor desempenho para uma atividade já conhecida, no caso uma relação contextual entre agente e ambiente. O nível instintivo possui uma coleção de agentes, cujo conhecimento são regras de produção simbólica, que representam *esquemas motores* a serem selecionados sobre o nível reativo. Em resumo, cada agente instintivo codifica um esquema motor específico, que é acionado sob uma visão do contexto corrente.

A partir da codificação de esquemas nos agentes instintivos sobre o nível reativo, surge um novo problema: “*Como selecionar os agentes instante-a-instante*”? A questão é encaminhada a um seletor, que escolhe o agente mais apto ao contexto. Há um conjunto de agentes instintivos, cujo mapeamento interno (o conhecimento simbólico de cada agente) são os esquemas sob nível reativo [Ark93, Tur93, Tur95], de tal modo que o agente mais habilitado para o contexto responde com uma ação. Por exemplo, caso o AA se encontre frontalmente a uma parede, um esquema que dispare uma ação à frente será um insucesso por completo. Uma vez selecionado o agente mais habilitado a um contexto, esse possui o direito de ler outros dados perceptuais que lhe interessem, como a visão e ϕ . Enfatiza-se que essa escolha ocorre sobre o conjunto de agentes ativos ou presentes nesse nível.

O modelo de agente no nível instintivo é diferente dos agentes reativos. Aqui os agentes possuem um caráter puramente simbólico. Os agentes instintivos descrevem esquemas internos dos AA's, que não se encontram explícitos no circuito de agentes

¹¹Ver capítulo 7.

reativos por si só. O nível instintivo atua como um tipo de “*meta-conhecimento*” sobre o reativo. A funcionalidade do AA é dada por uma combinação coerente dos agentes ativos nesse nível. Contudo, o nível cognitivo é o responsável pela definição dos agentes presentes. A seguir, descrevem-se os principais agentes modelados nos protótipos (*Brooks*, *Steels* e *Maja*). Finalmente, os selecionadores de agente utilizados no nível instintivo são formalizados.

5.6.1 Agentes Instintivos

Os agentes instintivos a serem descritos competem para uma habilitação de um esquema motor sobre o nível reativo, onde apenas um só é o escolhido, segundo uma proximidade contextual. As entradas sensoriais versus o seu contexto espacial definem o seu aspecto *comportamental*. Uma visão comportamentalista é usada para realizar as descrições dos agentes. A idéia é estereotipar um comportamento apropriado segundo um contexto espacial corrente. Obviamente, a idéia leva a uma explosão de combinações, pois o AA se desloca em ambientes desconhecidos, inerentemente infinitos, gerando um comprometimento do número de contextos. Segundo o modelo descrito para o agente, sem memória e com sensores primários, há uma existência infinita de contextos. O problema já foi levantado igualmente por R. Turner em [Tur93, Tur95].

O impasse da explosão de combinatorial é contornado utilizando-se da idéia de *proximidade ambiental*, conduzindo-se, também, a um *comportamento aproximado*. Com base no modelo físico do AA (apresentado no início do capítulo), um conjunto típico de situações pode ser identificado, através de seus cones sensoriais. O conjunto representativo de ambientes qualifica a grande maioria de instâncias detectadas pelas habilidades físicas do AA. Por exemplo, se o cone esquerdo e o frontal apresentarem baixos valores, isso implica que o agente se encontra num canto de sala fechado a sua frente e à esquerda. Um comportamento possível a ser instanciado é o agente retroceder ou girar à direita que está livre. Ou seja, os comportamentos são disparados segundo limiares internos dos agentes instintivos, que são selecionados segundo uma proximidade contextual. O agente que possuir os limiares de disparo mais próximos ao contexto apresenta uma maior chance de acionar um esquema sobre o nível reativo. Sendo assim, há uma faixa de limiares internos aos agentes instintivos, que caracterizam possíveis situações no ambiente.

De imediato surge a questão: “*Como valorar os limiares de disparo de um agente instintivo, segundo um número ilimitado de contextos?*” Neste trabalho utilizou-se o senso comum. Outros métodos mais sofisticados (e de elevado custo), como os da computação evolutiva poderiam ser usados, mas como a idéia do senso comum funcionou bem num primeiro momento, preferiu-se manter essa estratégia. Vale salientar que a determinação ótima dos limiares é um trabalho de grande esforço computacional.

A leitura nos cones sensoriais do AA é dada pela tripla $(x_{esq}, x_{dir}, x_{fre})$, que são relativos a um contexto corrente. Contudo, cada agente instintivo possui o seu limiar característico interno, que é dado por: $(l_{esq}, l_{dir}, l_{fre})$, acrescidos de variações

percentuais acima e abaixo dos originais, com os quais se busca um casamento aproximado com os valores correntes $(x_{esq}, x_{dir}, x_{fre})$. De modo análogo aos sistemas nebulosos, há uma curva de pertinência aproximada sobre cada um dos valores $(l_{esq}, l_{dir}, l_{fre})$. Os valores $(l_{esq}, l_{dir}, l_{fre})$ são definidos no nível cognitivo, podendo-se, portanto, modificar as particularidades de disparo para cada agente. Em resumo, busca-se identificar a proximidade existente entre os valores lidos pelos sensores $(x_{esq}, x_{dir}, x_{fre})$ e qual dos agentes instintivos reage melhor a um dado contexto ambiental.

Com base nas considerações acima e demais sensores do modelo físico do AA, define-se um agente instintivo genérico por:

$$Agente_k = (l_{esq}^k, l_{dir}^k, l_{fre}^k, \varepsilon_k, A_k, visão, \phi) \quad (5.6)$$

Os valores *visão* e ϕ são entradas sensoriais disponíveis ao agente selecionado. Considerando um *k-ésimo* agente instintivo selecionado, habilitando um esquema ε_k no nível reativo e visando a realização de uma ação A_k . A ação desempenhada nesse nível conduz aos valores θ e d (ver expressão 5.4). Em resumo, os agentes instintivos são formalizados a partir da expressão 5.6, com variações de um termo ou outro. Observa-se, ainda, que as equações 5.6 e 5.1 são consistentes entre si, gerando os seguintes termos:

$$l_k : (x_{esq}, x_{dir}, x_{fre}) \times \varepsilon_k \rightarrow A_k \quad (5.7)$$

onde l_k é a locomoção produzida pelo *k-ésimo* agente.

Alguns agentes são comuns aos três protótipos, outros apresentaram ligeiras modificações nas implementações. Os valores de “visão” e ϕ , opcionalmente, estão sempre disponíveis aos agentes que necessitarem de tal conhecimento. A partir das equações 5.6 e 5.7, os agentes instintivos implementados nesta tese são descritos por:

1. *Agente maximiza à frente*: esse agente visa coordenar um esquema do nível reativo, considerando que o AA deseja ir à frente e o caminho está livre nessa direção. Uma vez detectado que um movimento à frente é possível, apenas os agentes que formam o seguinte esquema são válidos¹²:

$$\varepsilon = \{\#2, \#5, \#8\}$$

Os ramos da esquerda e da direita são desconsiderados, pois não interessam num movimento à frente. No caso, os valores que definem a ação são dados por:

$$\begin{aligned} \theta &= 0^0 \\ d &\leq 0.5 \cdot Amp \end{aligned}$$

¹²Relembrando que todos esquemas a seguir são definidos de acordo com a figura 5.7.

Isso é, não há variação angular para o próximo passo, que tem um valor máximo de aproximadamente 50% do comprimento¹³ de seu cone sensorial (“*Amp*” é a sua amplitude);

2. *Agente oráculo*: esse agente visa direcionar o AA a uma fonte atratora. A diferença angular entre o AA e a fonte é dada por ϕ , que tem seu valor disponível toda vez que o agente for escolhido. As fontes são do tipo de reabastecimento, sonora, uma luz a seguir, etc, de modo que o agente procura se alinhar em direção à fonte. Há uma habilidade de giro em torno de si, caso o AA não esteja apontando na direção do foco. O giro pode ocorrer pela esquerda ou pela direita, de modo que dois esquemas possíveis a serem controlados no nível reativo são dados por:

$$\begin{aligned}\varepsilon_{esq} &= \{\#1, \#2, \#4, \#5, \#7, \#8\} \\ \varepsilon_{dir} &= \{\#2, \#3, \#5, \#6, \#8, \#9\}\end{aligned}$$

Em resumo, o comportamento típico é orientar o cone frontal do AA a uma fonte, fazendo uma curva ou à esquerda ou à direita. A escolha é feita pelo menor ângulo de desvio. Caso $0^0 \leq \phi \leq 180^0$, o giro é pela esquerda (sentido anti-horário), caso contrário, se $-180^0 \leq \phi \leq 0^0$, o giro é pela direita. Portanto, o valor final da ação é dada por:

$$\begin{aligned}\theta &= \phi \\ d &\leq 0.2 \cdot Amp\end{aligned}$$

O valor de $0.2 \cdot Amp$ estabelece um pequeno deslocamento do AA, enquanto o mesmo realiza a rotação em direção à fonte atratora;

3. *Agente arrasta-se à esquerda*: esse agente considera os feixes esquerdo e frontal como referenciais em sua navegação. O comportamento do agente é caminhar à frente, considerando uma proximidade de uma parede à esquerda. O esquema considerado no nível reativo é dado por:

$$\varepsilon = \{\#1, \#2, \#4, \#5, \#7, \#8\} \quad (5.8)$$

A proposta dessas ativações parciais sobre circuitos físicos vem do trabalho de C.M. Comer e J.P. Dowd [CD93], onde uma barata foi submetida a um túnel de vento sob várias direções. Foi verificado que alguns circuitos neurais eram mais ativados que outros, em face à essa fonte de vento.

¹³Alguns dos valores aqui adotados são quase arbitrários, objetivando apenas uma navegação robusta e segura, sem comprometimentos com precisão.

Tendo em vista a forte restrição que é o deslocamento do AA junto a uma parede, a proposta é estabelecer uma função compensativa para o seu desvio angular, possibilitando um controle eficaz e de baixo custo. Nesse caso, o feixe esquerdo deve permanecer em contato com a parede, de modo que: “*quando o AA estiver muito próximo à parede, ele deve se afastar e, se estiver muito longe, deve se aproximar*”. O objetivo é tornar suave e eficaz o controle dos movimentos de se afastar ou de se aproximar das paredes. Para isso, utilizou-se uma função senoidal, com um ponto médio de 50% do comprimento do sonar esquerdo.

A função senoidal compensativa aqui sugerida considera 50% da amplitude de um feixe como ponto de equilíbrio e o esquema 5.8, resultando num valor à ação motora da equação 5.4, a qual é dada por:

$$\theta = \arcsin(2 \cdot (x_{esq} - 0.5)) \quad (5.9)$$

$$d \leq 0.5 \cdot Amp \quad (5.10)$$

Em resumo, a expressão senoidal acima visa manter o AA com um alinhamento paralelo em relação à parede, quando o sinal do cone sensorial esquerdo estiver operando em aproximadamente 50% de sua capacidade. O valor de 50% em 5.9 está de certa forma relacionado com o deslocamento “d”, pois um grande valor no deslocamento “d” em direção oposta à parede faria com que o AA perdesse o contato com a parede, o que seria uma situação indesejável. O valor de 50% é apenas um critério arbitrário de segurança, implicando em ter o cone sensorial com 50% de sua captação junto à parede esquerdo;

4. *Agente arrasta-se à direita:* esse agente considera os feixes direito e frontal como referenciais em sua navegação. O comportamento do AA é caminhar à frente considerando uma proximidade de uma parede à direita. O esquema considerado no nível reativo é dado por:

$$\varepsilon = \{\#2, \#3, \#5, \#6, \#8, \#9\} \quad (5.11)$$

Valendo-se de todas considerações do item anterior, a ação motora calculada a partir do esquema 5.11 é dada por:

$$\theta = \arcsin(2 \cdot (0.5 - x_{dir})) \quad (5.12)$$

$$d \leq 0.5 \cdot Amp \quad (5.13)$$

As expressões senoidais desse agente e do anterior funcionam como uma função compensatória ativa, similar ao movimento de um pêndulo. Em [Ste94d], L.

Steels utiliza-se de curvas compensatórias tipo retas (portanto, mais simples que a função atribuída a θ) para descrever o diagrama de espaço-fase¹⁴ da translação que ocorre entre ativar um motor esquerdo, seguido pelo direito;

5. *Agente de becos*: esse agente visa livrar o AA de becos ou equivalentes o mais rapidamente possível. A detecção é feita considerando as entradas dos três cones sensoriais ($x_{esq}, x_{dir}, x_{fre}$), cujos valores estejam próximos a zero. Enfim, detecta-se um caminho fechado à frente e aos lados. A idéia é ativar um esquema que gire o AA à esquerda ou à direita, uma escolha aleatória, objetivando uma maximização do cone frontal. Um desses dois esquemas são habilitados para sair dessa situação de equilíbrio:

$$\begin{aligned}\varepsilon_{esq} &= \{\#1, \#4, \#7\} \\ \varepsilon_{dir} &= \{\#3, \#6, \#9\}\end{aligned}$$

Portanto, há uma rotação (um giro) do AA. Quanto a sua ação desencadeada, seus valores finais resultam em:

$$\begin{aligned}\theta &= 180^\circ \\ d &= 0.3 \cdot Amp\end{aligned}$$

A ação comportamental acima descreve um retorno no sentido contrário e um passo de 30% da amplitude do cone frontal. As ações escolhidas são simples e procuram evitar conflitos indesejáveis;

6. *Agente conflito sem orientação*: esse agente resolve um típico problema da IA: o *conflito*. Nesse caso, o AA encontra-se frontalmente a uma parede, e os cones sensoriais esquerdo e direito possuem os mesmos valores. Um equilíbrio que provoca um conflito. Análogo ao agente de becos, não há uma consulta sobre os valores de “visão” e ϕ . Novamente, um esquema que gire o AA à esquerda ou à direita é habilitado, de modo que:

$$\begin{aligned}\varepsilon_{esq} &= \{\#1, \#4, \#7\} \\ \varepsilon_{dir} &= \{\#3, \#6, \#9\}\end{aligned}$$

Vale lembrar que os lados do AA estão livres, portanto a ação emprega um dos dois esquemas acima. Novamente, há uma escolha aleatória de qual esquema habilitar. Os valores finais dessa ação resultam em:

¹⁴Um diagrama de espaço-fase mostra a evolução da dinâmica de todos os estados possíveis de um sistema.

$$\begin{aligned}
\theta &= +90^0 && (\text{usando } \varepsilon_{esq}) \text{ ou} \\
\theta &= -90^0 && (\text{usando } \varepsilon_{dir}) \\
d &= 0.05 \cdot Amp
\end{aligned}$$

Após a ação, o agente se posicionará paralelamente à parede. Novamente, buscou-se uma solução de simples implementação, de modo que o deslocamento seja mínimo em 5% da amplitude, com o simples objetivo de sair da situação de equilíbrio;

7. *Agente conflito com orientação*: situação análoga à anterior, exceto que o desvio angular do AA com a marca corrente ϕ é consultado¹⁵, visando determinar para qual dos lados o AA vai girar. Sendo assim, tem-se:

$$\begin{aligned}
\varepsilon_{esq} &= \{\#1, \#4, \#7\} && \text{se } 0^0 < \phi \leq 90^0 \\
\varepsilon_{dir} &= \{\#3, \#6, \#9\} && \text{se } -90^0 \leq \phi < 0^0
\end{aligned}$$

Caso $\phi = 0^0$, nenhum desvio entre a linha mediana do cone central e a marca, há uma escolha aleatória. Quanto à ação resultante tem-se:

$$\begin{aligned}
\theta &= +90^0 && (\text{usando } \varepsilon_{esq}) \text{ ou} \\
\theta &= -90^0 && (\text{usando } \varepsilon_{dir}) \\
d &= 0.05 \cdot Amp
\end{aligned}$$

Um deslocamento mínimo de 5% da amplitude do cone é sugestivo pela “quebra” de equilíbrio nessa situação de conflito.

8. *Agente para um canto à esquerda*: dado um comportamento de se movimentar junto à parede pelo lado esquerdo, eventualmente se depara com obstáculos frontais típicos, como cantos de sala. O esquema utilizado para contornar um canto côncavo é dado por:

$$\varepsilon = \{\#1, \#2, \#4, \#5, \#7, \#8\}$$

Nesse momento, esse agente produz sucessivas rotações para θ e “ d ” até sair do canto à esquerda:

$$\begin{aligned}
\theta &= -15^0 && (\text{sucessivos giros à direita}) \\
d &= 0.05 \cdot Amp
\end{aligned}$$

¹⁵Lembrando que o valor de ϕ é disponível aos agentes que desejarem fazer uma consulta ao sensor com esse dado.

Novamente, uma “heurística” foi escolhida de forma elementar (simplista). Ao final de aproximadamente 6 (seis) avanços frontais, o AA teria realizado um giro de -90^0 (6×-15^0), onde o comportamento resultante é o desvio frontal de um obstáculo;

9. *Agente para um canto à direita*: análogo ao agente anterior, exceto que o comportamento do AA é junto à parede direita, que eventualmente se depara com obstáculos frontais. Nesse caso, o esquema habilitado é dado por:

$$\varepsilon = \{\#2, \#3, \#5, \#6, \#8, \#9\}$$

Nesse momento, esse agente produz sucessivas rotações para θ e “ d ” até sair do canto à direita:

$$\begin{aligned}\theta &= +15^0 && (\text{sucessivos giros à esquerda}) \\ d &= 0.05 \cdot Amp\end{aligned}$$

Considerando que o AA realize aproximadamente 6 (seis) rotações 90^0 (6×15^0), há, ainda, uma distância final que garante um certo afastamento da parede. O valor de “ $0.05 \cdot Amp$ ” corresponde a 5% do comprimento do cone, portanto 6 (seis) avanços frontais equivalem a manter uma margem de 70% da amplitude afastada da parede;

10. *Agente “última chance”*: Caso os seletores¹⁶ de agentes instintivos falhem, o nível reativo inteiro é habilitado, de modo que o esquema é dado por:

$$\varepsilon = \{\#1, \dots, \#9\}$$

Os valores resultantes da ação θ e “ d ” seguem uma composição de forças baseadas na 2a. lei de Newton. Contudo, o deslocamento é limitado por:

$$d \leq 0.5 \cdot Amp \quad (\text{Critério de segurança, 50\% da amplitude.})$$

Esse agente foi um precursor na concepção desta tese. Inicialmente só havia a camada reativa, que tornava ativo todos os agentes nebulosos. Alguns resultados da navegação desse agente são encontrados em [dSGB096].

Algumas considerações sobre escolhas das ações produzidas pelos agentes acima descritos precisam ser feitas:

¹⁶Descritos na seção seguinte.

- Em momento algum, o requisito de movimentos precisos e perfeitos foram determinantes. O critério de um movimento aceitável e seguro, sem posicionamentos muito próximos a obstáculos, foi considerado como satisfatório;
- A heurística da escolha pelo mais simples foi relevante. A exemplo dos efeitos da natureza, que sempre tomam caminhos e realizam escolhas, segundo critérios de simplicidade;
- A junção de componentes elementares que levam a ações não-previsíveis ou não-lineares, são considerados como *comportamentos emergentes*, segundo critérios de S. Forrest [For90]. A arquitetura do AA aqui concebida apresenta muitos agentes internos, que competem e cooperam entre si, sem nenhum tipo de controle centralizador e/ou política interna de escalonamento, tornando esta arquitetura com conceitos semelhantes ao de *comportamento emergente* [For90].

5.6.2 Seletores de Agentes Instintivos

Os agentes desse nível entram num processo competitivo. Portanto, necessita-se conhecer o agente selecionado. O agente selecionado tem o direito de resposta, podendo consultar os valores de “visão” e ϕ . A seleção visa o agente mais ajustado ou próximo aos requisitos do contexto das entradas sensoriais. A existência de um seletor de agentes é comum aos selecionadores de ações em projetos de AA’s. Esse ponto foi comentado por P. Maes em [Mae91c]. As granularidades são diferentes, mas a natureza do problema é a mesma.

O processo de escolha do agente mais ajustado é um dos problemas mais antigos na IA, isto é, o problema de casamento de padrões (“*pattern-matching*”). Para resolver esse problema, necessita-se de um selecionador de agentes que apresente alguns requisitos de projeto, como:

- Ser flexível (ser de fácil alteração a particularidades do projeto);
- Ter um bom grau de precisão (boa sensibilidade);
- Ser de baixo custo de implementação computacional;
- Ser de baixo custo operacional. Por exemplo, eliminar a etapa de treinamento de padrões;
- Ter uma bom desempenho computacional, isto é, ser veloz.

O selecionador de agentes procura um “bom casamento” entre os sinais sensoriais correntes e os *limiares de disparos* de cada agente. Cada agente possui um conjunto característico de limiares de disparos, que levam a uma ação típica num dado contexto. Um comportamento é ativado cada vez que as condições ambientais forem favoráveis. Por exemplo, se o AA detectar que se encontra em um beco sem saída, dispara uma seqüência de ações que permitem um retorno à entrada.

O problema reside em disparar o agente com a ação mais apropriada numa dada situação. O objetivo do selecionador é escolher o agente que melhor se adapte a uma situação. A seleção acontece sobre a lista de agentes ativos no nível instintivo. Uma boa escolha é baseada em uma das muitas definições de comportamento: “*uma seqüência de ações bem-sucedidas sobre um dado contexto, que consigam caracterizar uma regularidade*”.

O problema poderia ser encaminhado sob uma visão conexcionista, usando as redes neuronais. Outra opção é a evolucionista, com os algoritmos genéticos ou, ainda, a probabilista, com o teorema de Bayes. Contudo, essas alternativas contrariam alguns dos requisitos de projetos citados acima.

A idéia que “*o melhor casamento é o de menor diferença entre os pares*” estimulou a construção de um primeiro seletor de agentes, resultando num *seletor diferencial*, cujo funcionamento é primário e impreciso, embora tenha atendido às primeiras necessidades de uma implementação computacional. Em seguida, optou-se por um seletor mais sofisticado baseado na lógica nebulosa. Os protótipos estão operando com o *selecionador nebuloso* numa primeira instância. Quando esse falha, não conseguindo instanciar nenhum agente, o *selecionador diferencial* é acionado. Apesar da imprecisão, o diferencial sempre retorna a escolha de um agente. Em seguida é descrita a implementação desses dois seletores de agentes.

Selecionador Diferencial:

O método tem por base a menor diferença entre uma situação corrente ($x_{esq}, x_{dir}, x_{fre}$) e os limiares de disparo ($l_{esq}, l_{dir}, l_{fre}$) de cada agente. A diferença é absoluta, portanto, o método não é muito preciso para um grande número de entradas. Como nessa instância consideram-se apenas três valores, o método tem uma eficiência aceitável.

A implementação é descrita a seguir. Seja CI o conjunto de agentes ativos¹⁷ no nível instintivo, portanto, $CI = \{ag_1, ag_2, \dots, ag_m\}$. Cada agente ag_i é definido pela seguinte estrutura de dados:

$$ag_i = [i, l_{esq}, l_{dir}, l_{fre}] \quad (5.14)$$

onde i é o número identificador do agente e l são os limiares de disparo dos sensores laterais e frontal. Seja $X_{atual} = [x_{esq}, x_{dir}, x_{fre}]$, os valores lidos através dos sensores, respectivamente, dos lados esquerdo, direito e frontal.

O critério das diferenças absolutas é aplicado para cada agente, sendo definido por:

$$dif_ag_i = abs(l_{esq} - x_{esq}) + abs(l_{dir} - x_{dir}) + abs(l_{fre} - x_{fre})$$

¹⁷Dos 10 agentes instintivos definidos anteriormente, apenas alguns estarão habilitados para uma funcionalidade desejada, chamados de *ativos*.

Esse cálculo é feito para cada agente, obtendo-se uma lista de diferenças absolutas, para todos os agentes. Numa segunda etapa, o agente selecionado é o de menor diferença da lista, dado por:

$$ag_selec = \min(dif_ag, dif_ag_2, \dots, dif_ag_m)$$

Uma vez selecionado o agente, uma ação característica é disparada. Em resumo, esse método parte da premissa que o “*melhor casamento de padrões ocorre entre objetos de menor diferença numérica entre si*”.

Selecionador Nebuloso:

Este seletor é baseado no raciocínio da lógica nebulosa [Lee90a, Lee90b]. Portanto, há um esquema de raciocínio nesse seletor usando um método inferencial. O método inferencial de raciocínio aqui adotado é o *Modus Ponens* nebuloso [TS87], definido por:

$$\frac{A \rightarrow B \quad A'}{B'}$$

No processo inferencial acima, tanto na fuzzificação como na desfuzzificação, os operadores adotados foram “mín” e “máx”, respectivamente [Sug85, FK94, Miz95].

A entrada de cada sonar é um número real entre 0 e 1, onde valores próximos a 0 tem uma semântica do tipo “*obstáculos muito próximos*”, e os valores próximos a 1 indicam que “*obstáculos estão afastados*”. Um cone sensorial com valor 0 indica que ele se encontra sobre um obstáculo. Do contrário, o valor 1 indica que não há obstáculos nas vizinhanças. Para os três sonares do AA tem-se:

$$\begin{aligned} 0 &\leq x_{esq} \leq 1 \\ 0 &\leq x_{dir} \leq 1 \\ 0 &\leq x_{dir} \leq 1 \end{aligned}$$

Os “*m*” agentes ativos são definidos individualmente por seus limiares de disparos e um número identificador. Essa estrutura é dada por:

$$ag_i = [i, l_{esq}, l_{dir}, l_{fre}]$$

O estado corrente de leitura dos sensores é definido por:

$$X_{atual} = [x_{esq}, x_{dir}, x_{fre}]$$

A base nebulosa de definição dos “*m*” agentes é dada por um conjunto de regras do tipo:

$$\text{se } (X_{esq} = X_{1i}) \text{ e } (X_{dir} = X_{2i}) \text{ e } (X_{fre} = X_{3i}) \text{ então } (Ag_selec = I) \quad (5.15)$$

tal que $i = 1, \dots, m$.

O conjunto de regras formado a partir da expressão 5.15 possui um equivalente neuronal, cuja representação resumida é dada na figura 5.9. A construção de uma rede neuronal equivalente pode ser encontrada em [Wak93]. Detalhes adicionais, ver [TG96], onde a equivalência é estendida para uma rede neuronal do tipo “*feedforward*”, como o exemplo da figura 5.9.

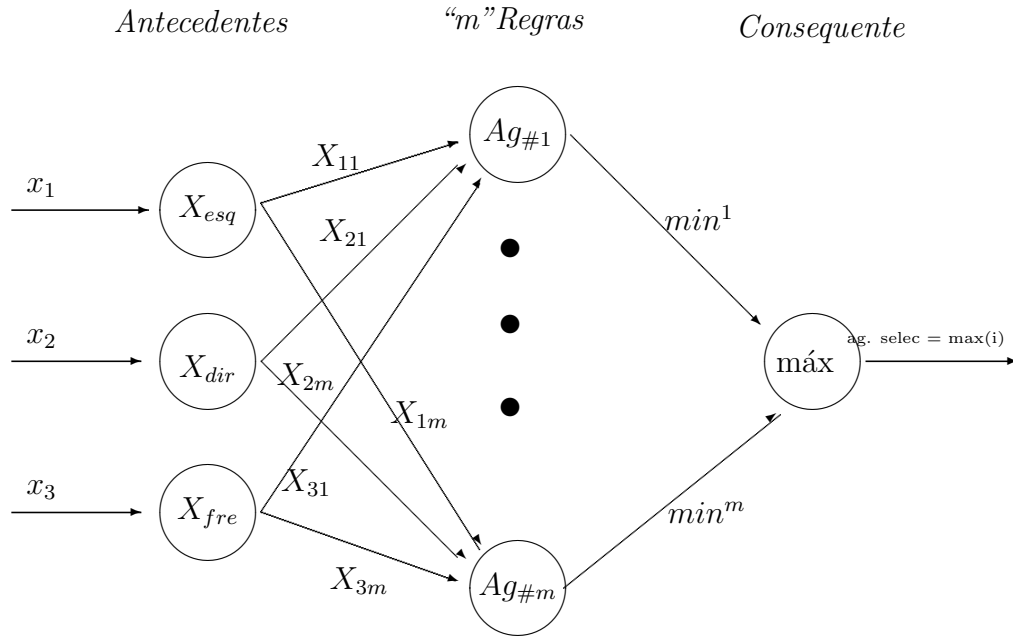


Figura 5.9: Rede Neural Equivalente ao Conjunto de Regras de Produção

As funções de pertinência que caracterizam cada regra acima, junto a seus limiares, são do tipo triangular. A razão dessa escolha é a simplicidade e a eficiência no tocante à sua sensibilidade. O detalhe é associar uma curva de pertinência do tipo triangular ao valor do limiar de disparo do cone. Por razões diretas, o valor de disparo é escolhido como sendo o centro do triângulo. A curva pertinência de um dos cones sensoriais é apresentada na figura 5.10.

O valor de pertinência é derivado a partir das seguintes equações:

$$\mu_{X_{e,d,f}}(x) = \begin{cases} 1 - \frac{|x-l|}{\delta} & \text{para } (l - \delta) < x < (l + \delta) \\ 0 & \text{para } x \leq (l - \delta); \quad x \geq (l + \delta) \end{cases}$$

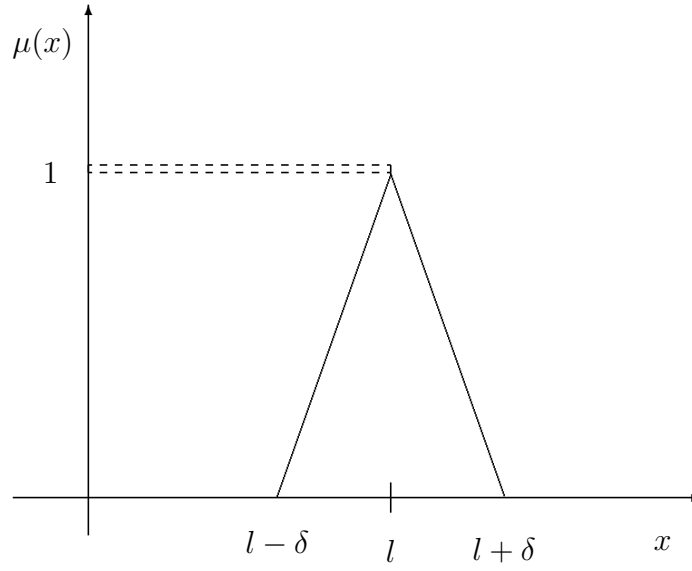


Figura 5.10: Curva Triangular de Pertinência

Considerando as premissas acima, os passos do procedimento do seletor nebuloso são abaixo descritos:

1. Fuzzificar a entrada corrente (leitura dos sonares) sobre as curvas triangulares que definem os limiares dos “m” agentes ativos;
2. Considerar todos agentes que apresentam $\mu_{X_{esq}}(x_1) \neq 0$; $\mu_{X_{dir}}(x_2) \neq 0$ e $\mu_{X_{fre}}(x_3) \neq 0$. Esse passo é o “casamento de padrões” da parte antecedente das regras. São válidos os “n” agentes que apresentarem tal interseção com o valor corrente, tal que $n \leq m$. Caso $n = 0$, saia do procedimento e chame o selecionador diferencial;
3. Aplicar o operador “mín” sobre a lista de agentes “n”, de modo que:

$$\text{mín}^i = \text{mín}(\mu_{X_{esq}}(x_{1i}), \mu_{X_{dir}}(x_{2i}), \mu_{X_{fre}}(x_{3i}))$$

onde $i = 1, \dots, n$;

4. Aplicar o operador “máx” sobre a lista dos “n” agentes candidatos. Isso é:

$$\text{máx}(i) = \text{máx}(\text{mín}^1, \text{mín}^2, \dots, \text{mín}^n)$$

5. Desfuzzificar a partir dos máximos, do passo anterior, obtendo com isso o índice do agente a ser escolhido pelo maior valor da lista de candidatos. Sendo assim:

$$ag_selec = i$$

O método acima ainda não apresenta as boas características classificatórias das redes neuronais. Contudo, o seu custo computacional, sua clareza e sua precisão o tornam um método aceitável às expectativas e aos requisitos da arquitetura proposta.

Uma vez selecionado um dos agentes ativos, esse dispara a sua ação de modo cooperante à camada reativa. Há vários agentes ativos no nível instintivo, o que os torna candidatos ao disparo de uma ação. Contudo, a definição dos agentes disponíveis no nível instintivo é uma tarefa do nível cognitivo.

5.7 Nível Cognitivo

O nível cognitivo descreve as habilidades do AA, na seleção dos agentes que estarão ativos no nível instintivo. Há uma analogia com as regiões do córtex cerebral, que cooperam e competem entre si, provendo o ser humano de alguma cognição. A partir do córtex cerebral estabelecem-se os movimentos voluntários, que foram induzidos por uma percepção sensorial ou uma montagem estereotipada de comportamentos. No córtex é que se localiza a estruturação de uma idéia, da montagem das linguagens, etc. Seguindo-se essa linha, o nível cognitivo apresentado por esta arquitetura também exhibe a definição de habilidades aos protótipos implementados.

Algumas propostas de B.H.-Roth [HR95] referem-se a comportamentos dirigidos a nichos. Apesar de restritos em suas atividades, apresentam uma boa robustez quando sujeitos a situações desconhecidas ou a imprevistos. A robustez é traduzida em termos da capacidade adaptação do agente em um ambiente [Bro91c, Bro91b].

As aplicações aqui desenvolvidas sobre esta arquitetura seguem a idéia de comportamentos dirigidos a nichos. Nos dois primeiros protótipos (*Brooks* e *Steels*), o nível cognitivo é modelado de modo elementar, utilizando-se de um arquivo-texto típico. Contudo, o mesmo não ocorre no último protótipo (*Maja*). Na arquitetura em questão, a funcionalidade é definida pela combinação coerente sobre os agentes do nível instintivo. Os agentes no nível instintivo podem ser conflitantes e afastarem-se de um propósito comum cooperante. Sendo assim, a habilitação ocorre segundo algumas restrições, de acordo com o tipo de agentes instintivos presentes. A escolha desses agentes é feita no nível cognitivo, definindo uma funcionalidade ao AA sobre um nicho. Tal tarefa é realizada via um arquivo-texto.

Finalmente, retoma-se ainda a questão de definir o estados sob uma visão comportamental, onde a variável E (conjunto de estados) da expressão 5.1 é definida a partir do comportamento funcional do protótipo. Avaliando-se experimentalmente o comportamento de cada protótipo, gera-se um *autômato comportamental* (AC) típico para cada AA, os quais são apresentados ao final das subseções. O AC é definido a partir de um autômato de estado finito do tipo máquinas de Moore [Arb69]. As máquinas de Moore apresentam saídas que dependem de um estado e da entrada corrente. O arco de transição entre dois estados de uma máquina de Moore possui

o símbolo reconhecido e uma saída gerada. Como já discutido, os estados caracterizam os comportamentos típicos ou regulares do AA, e suas transições ocorrem em função do estado corrente e de uma entrada atípica e/ou de uma motivação interna. Em M.Roisenberg e outros [Aze96], comportamentos reativos de um AA são modelados via uma máquina de estados finitos, implementada por uma rede neuronal recorrente, que foi utilizada para prover o sistema com memória e dinamismo de seu comportamento. O espaço de navegação do agente é do tipo reticulado.

5.7.1 Brooks

O protótipo Brooks visa encontrar a saída de um labirinto, utilizando-se de um dos seus sensores laterais. A idéia é caminhar próximo à parede, de modo que não se perca contacto com a mesma e não se permita uma troca de referencial. A heurística é simples, mas que sempre gera um caminho entre a entrada e a saída. O termo “referencial” apresenta dois sentidos:

- Realizar um giro em torno de si e tomar o caminho ao contrário do inicial. Nesse caso, o referencial do sentido original de caminhar é perdido;
- Perder o contacto de uma parede e assumir a outra parede como referencial.

Para que isso não ocorra, os agentes instintivos de caminhar junto às paredes esquerda e direita não podem estar ativos simultaneamente. A perda de contacto por um momento de um dos lados colocaria uma ativação competitiva por parte dos dois agentes.

Dentro do nível cognitivo, habilitam-se agentes instintivos e definem-se seus limites de disparo, tudo isto visando uma coerência funcional. Tais definições são feitas com um arquivo-texto. Por exemplo, no caso de caminhar tendo-se como referencial a parede esquerda, um possível arquivo-texto é dado por:

```
"XA" 1 0.99 0.99 0.99 ==> o AA usa o agt oraculo
"XA" 2 0.99 0.99 0.5 ==> o AA usa o agt de maximizar a frente
"HA" 5 0.45 0.99 0.85 ==> o AA arrasta-se pela esquerda (relac. #11)
"XA" 6 0.99 0.45 0.85 ==> o AA arrasta-se pela direita (relac. #12)
"HA" 7 0.13 0.13 0.13 ==> o AA encontra-se num beco sem saida
"XA" 8 0.9 0.9 0.43 ==> o AA encontra uma parede a frente: conflito
"XA" 9 0.9 0.9 0.55 ==> um agt p/a conflito c/ orientacao da marca
"HA" 11 0.35 0.99 0.45 ==> o AA aproxima-se num canto a esquerda
"XA" 12 0.99 0.35 0.45 ==> o AA aproxima-se num canto a direita
```

As alterações desse arquivo-texto são realizadas com um editor qualquer de texto. Os campos do arquivo-texto possuem as seguintes definições:

- O primeiro campo “HA” ou “ha” define que o agente se encontra *habilitado*. Qualquer outra definição desabilita tal agente. No exemplo acima, os agentes com “XA” em seus primeiros campos encontram-se inativos ou desabilitados;
- O segundo campo é um numeral que identifica o agente ativo no nível interno de programa. A identificação é pertinente ao simulador e não coincide com a ordem que foi apresentada anteriormente;
- Os três campos seguintes definem os limiares de disparo de cada sonar (esquerdo, direito e frontal, nessa ordem) do agente, isto é, $(l_{esq}, l_{dir}, l_{fre})$;
- Os trinta caracteres seguintes são usados para descrever textualmente o que faz cada agente instintivo. Esse campo de comentários não assume nenhum efeito sobre o programa. Serve exclusivamente para uma descrição textual do agente.

Observa-se que com apenas três agentes habilitados, consegue-se definir uma aptidão para esta arquitetura. No exemplo acima, caminhar arrastando-se próximo à parede esquerda é coordenado pelos agentes: “*arrastar-se à esquerda*”, “*contornar cantos à esquerda*” e “*sair de um beco*”. Ressalta-se que o limiar de disparo do agente “5” (“*arrastar-se à esquerda*”) tem o valor igual a 0.5 para o feixe esquerdo. Isso significa manter o AA a uma distância de aproximadamente 50% da capacidade sensorial do disparo do sonar. Vale salientar que essa atribuição é do limiar de disparo do agente e não se relaciona com a função compensatória interna, como descrito na seção do nível instintivo.

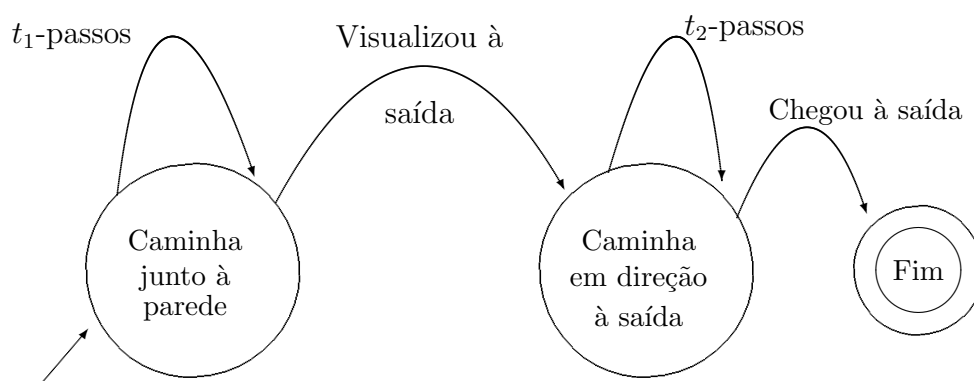


Figura 5.11: Nível Cognitivo - Brooks

Finalmente, o AC resultante da funcionalidade de Brooks é apresentado na figura 5.11. Inicialmente, o AA encontra-se próximo a uma parede e seu estado é definido pelo comportamento de caminhar rente a ela. O AA caminha junto à parede até avistar a porta de saída. Há uma permanência nesse estado por t_1 -passos. No momento que o AA visualiza a saída (é um evento atípico), um novo comportamento de caminhar nessa direção é caracterizado. A duração no estado é de t_2 -passos. Uma vez que o AA tenha alcançado a porta, seu ponto final, uma transição ocorre para um estado de término ou de parada.

5.7.2 Steels

O protótipo Steels possui a habilidade de encontrar objetos ou fontes atratoras, dispostos em um ambiente desconhecido. Os ambientes típicos são restritos a salas com objetos em seu interior, por exemplo, um laboratório, um escritório, uma sala, etc. Os obstáculos em seu interior são objetos como cadeiras, mesas, armários, birôs, etc, espalhados no ambiente. As fontes atratoras são definidas na simulação e formam uma lista de objetivos ou marcas que o AA deve alcançar. A partir do sinal recebido de uma fonte de uma atratora, o AA visa percorrer a lista pré-definida, na ordem em que foi criada. Sendo assim, o primeiro objeto instanciado é o primeiro a ser encontrado.

Nesse protótipo, o nível cognitivo também é definido pela ativação combinada sobre os agentes instintivos. Para o protótipo *Steels*, a definição do arquivo-texto é dada abaixo:

```
"HA" 1 0.99 0.99 0.99 ==> o AA usa o agt. oraculo
"XA" 2 0.99 0.99 0.5 ==> um agt para maximizar a frente
"HA" 5 0.5 0.99 0.85 ==> o AA arrasta-se pela esquerda
"HA" 6 0.99 0.5 0.85 ==> o AA arrasta-se pela direita
"HA" 7 0.13 0.13 0.13 ==> o AA encontra-se num beco sem saida
"HA" 8 0.9 0.9 0.33 ==> o AA encontra-se em conflito (perdido)
"HA" 9 0.9 0.9 0.55 ==> um agt para conflito com orientacao da marca
"HA" 11 0.35 0.99 0.35 ==> o AA encontra-se num canto a esquerda
"HA" 12 0.99 0.35 0.35 ==> o AA encontra-se num canto a direita
```

De imediato, observa-se um número muito maior de agentes habilitados (“HA” ou “ha”). Isto significa que o AA não possui restrições de navegação, tornando mais flexível e ampla a competitividade entre os agentes do nível instintivo. A competitividade reflete um comportamento dinâmico, oportunístico e mais adequado a situações ambientais imprevisíveis. Considerando que nenhum esquema de memória é aplicado à arquitetura, toda situação apresentada ao AA é vista como um novo contexto.

Salienta-se que ao contrário do protótipo *Brooks*, as habilidades de contornar objetos pelo lado esquerdo e direito estão simultaneamente ativos. São agentes que competem segundo uma sensibilidade contextual.

Segundo os resultados a serem apresentados no capítulo 6, o protótipo Steels apresenta algumas restrições de navegação. Dependendo do formato dos objetos no ambiente, eventualmente, o AA pode não conseguir contorná-los. Exemplifica-se com becos, salas recorrentes e outros objetos acentuadamente côncavos. A limitação deste protótipo motivou um incremento no nível de cognição, gerando o protótipo *Maja*.

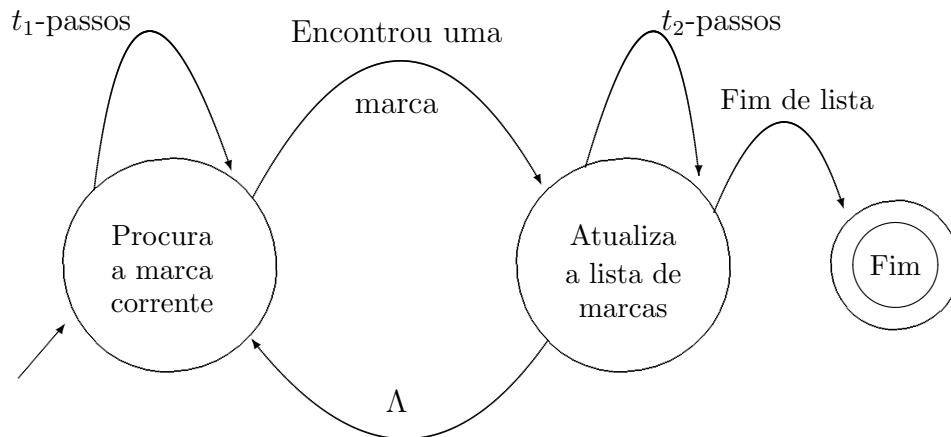


Figura 5.12: Nível Cognitivo - Steels

Retomando-se a questão dos estados, o AC resultante da funcionalidade de Steels é sumarizado na figura 5.12. De qualquer posição inicial do AA no ambiente, ele imediatamente visa caminhar em direção à primeira marca da lista, com duração de t_1 -passos. A cada marca encontrada, um novo estado é caracterizado, ocorrendo uma atualização de marcas. O período de permanência no estado de definir ou atualizar o AA para uma próxima meta tem uma duração de t_2 -passos. Sempre que existir uma marca pendente, o AA busca encontrá-la. Uma vez que o AA tenha percorrido todas as marcas definidas na lista, uma transição vazia é disparada pelo evento fim de lista, no qual o AA segue para um estado de parada ou de término.

5.7.3 Maja

O protótipo Steels apresenta limitações quanto à navegação em ambientes intrincados, como salas dentro de salas, etc. Tal fato é constatado pela completa ausência de memória do AA em locais já visitados e pelos tipos de sensores modelados. Para contornar tais limitações, o protótipo Maja é incrementado por um autômato comportamental no nível cognitivo, que gerencia os estados do AA. O autômato desenvolvido foi inspirado sob uma visão de um comportamento biológico de um inseto primário

tipo uma mosca. Observa-se que insetos não possuem um instinto aguçado para memorização de territórios. Por exemplo, ao objetivar uma saída por uma janela de vidro, a mosca está sendo atraída pelo odor ou pela luminosidade externa. Deparando-se com o vidro, a mosca fica numa investigação próxima à janela, de forma intermitente. Eventualmente, os sensores de odor da mosca vão se saturando, levando-a a uma desistência temporária de seu objetivo. Nesse período, ela voa para longe, buscando uma nova saída ou desistindo (talvez, esquecendo) do seu objetivo por um período de tempo. Decorrido um tempo, a mosca voa para outras direções, e pode retornar à janela. Verifica-se a existência de um ciclo que se repete, caracterizando estados como: “*voa em direção ao seu objetivo*”, “*satura ou se cansa do mesmo*” e “*voa para longe buscando alternativas*”.

O comportamento com base no odor é muito diferente para os mamíferos em geral, usado para reconhecer territórios, outros animais, épocas de acasalamentos pela liberação de feromônios por parte das fêmeas, etc. Por intermédio de instintos ocorrem processos como a *demarcação de territórios*, reconhecimento de locais já visitados, presas e predadores, tudo com uma precisão inata do animal. O exemplo do comportamento da mosca sob uma janela de vidro, provavelmente não seria repetido por um rato. Para os mamíferos, há um esquema de memorização mais sofisticado que envolve entradas sensoriais do tipo olfato, visão, gustação, audição, etc. Em outras palavras, uma modelagem comportamental da mosca é mais simples que a de um mamífero.

Considerando o comportamento do inseto acima, o nível cognitivo foi construído com base na intermitência de alguns insetos sob certos alvos, no caso, os voos feitos por uma mosca. Em alguns momentos, ela fica se tocando no vidro várias vezes, ora voa distante, até que finalmente uma abertura seja encontrada. A característica de ser intinerante e eventual a seu objetivo motivou a construção de um *autômato comportamental* (AC), com base nos três estados *temporais* descritos acima: “*segue em direção ao objetivo*”, “*satura ou se cansa junto à janela*” e “*voa para outras direções*”. Há um ciclo entre esses três estados. Adaptando-se esse ciclo ao caso do AA, um AC é construído conforme a figura 5.13. O AC da figura 5.13 passa a “*controlar*” os estados a *longo prazo* em que o AA se encontra. Com isso, o AC representa uma “*estratégia cognitiva*” para os comportamentos a *longo prazo* do protótipo Maja.

Os estados principais (círculos maiores) da figura 5.13 representam o comportamento típico do agente durante um certo tempo. Os arcos refletem as saídas produzidas a partir desse estado, que são eventos de disparo gerados a partir da superação de algum limiar no estado corrente, no caso um tempo de permanência. Os estados secundários (círculos menores) são aqueles que estão embutidos nos estados principais e que apresentam apenas características funcionais do modelo cognitivo. De acordo com a figura 5.13, os estados secundários são definidos por:

AM : Atualiza a marca. Uma vez que uma marca é encontrada no ambiente, precisa-se tomar um novo elemento da lista de marcas. Há uma estrutura de fila, tal que a primeira marca no ambiente é primeira a ser buscada. O símbolo “Λ”, a partir desse estado, significa uma transição espontânea, isto é, após a marca

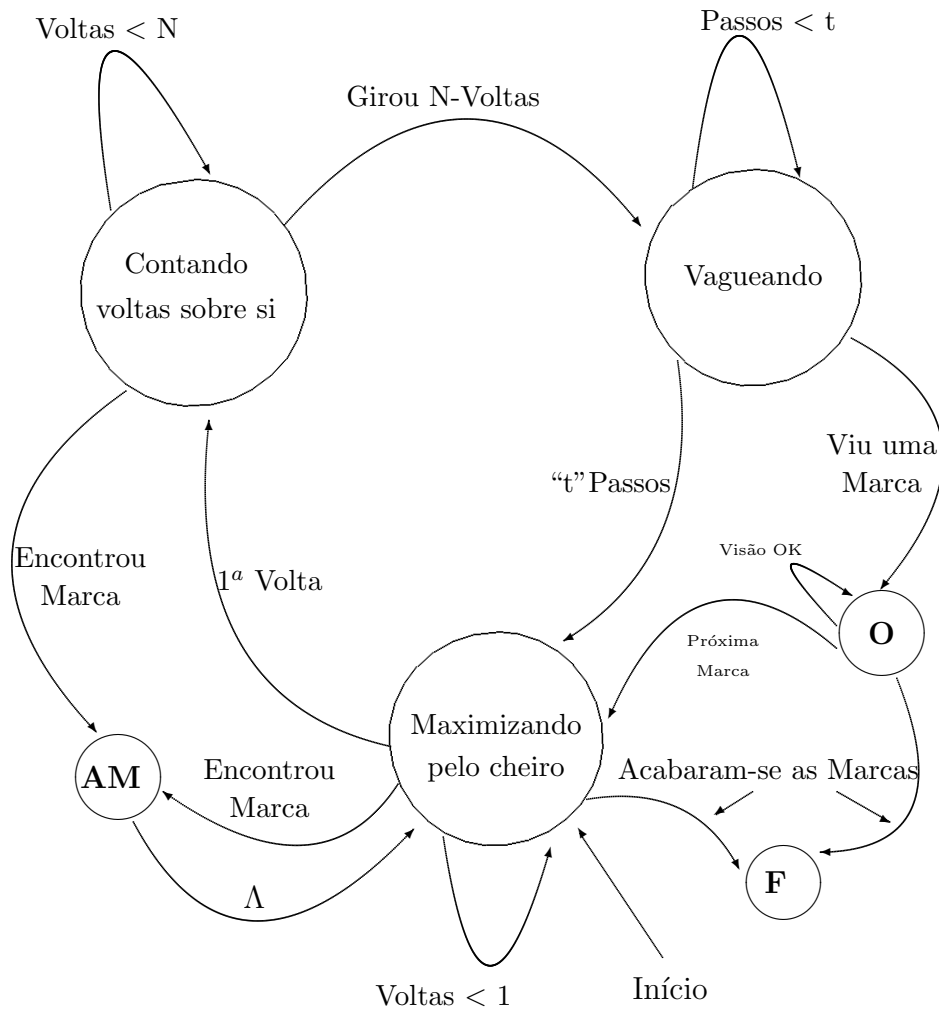


Figura 5.13: Nível Cognitivo - Maja

atualizada, o controle segue para o estado seguinte;

O : Consulta a um agente oracular. Eventualmente, se o AA se depara visualmente com a marca desejada, ele dispara os mecanismos de um agente oráculo, cujo objetivo é caminhar em direção à marca, conforme descrito anteriormente;

F : Fim. No momento em que as marcas acabaram, o AA cessa a sua busca.

Para os três estados de comportamentos característicos do inseto junto à janela, os estados comportamentais identificados para o AA são os seguintes:

- **Maximizando pelo cheiro** (1): o comportamento aqui disparado é semelhante ao protótipo *Steels*. Ele visa maximizar o deslocamento em direção a uma fonte atrativa, que é equivalente ao inseto voar em direção ao seu objetivo;
- **Vagueando** (2): o comportamento aqui disparado é itinerante, sem rumo, sem propósito. Simboliza a desistência temporária do AA de sua meta. Nesse estado, o AA caminha a esmo, de forma aleatória. Seu controle interno é com base num relógio (no caso, um certo número de passos), que memoriza apenas o passo anterior. Há uma semelhança com processos Markovianos em que o passo futuro depende apenas do estado corrente, isto é, independe da história passada. A permanência no estado é reforçada pela ocorrência freqüente da ativação desse estado. Sob uma visão biológica, é como se o inseto voasse o mais distante possível cada vez que atingisse a saturação de seu sensor olfativo;
- **Contando voltas sobre si** (3): o comportamento aqui acionado é de tomar alguma “*consciência*” sobre suas ações. No início ele visava a marca corrente, mas sempre mantendo uma consciência sobre um possível giro em torno de si. Quando um giro ou uma primeira volta em torno de si foi detectada, um novo estado de contar o número de voltas em torno de si é acionado. O estado é um monitoramento sobre suas ações, no qual ele continua a busca da marca, mas controla um certo número de voltas “N” sobre si mesmo. Caso um número máximo seja atingido, um novo comportamento é ativado. No caso, ele segue ao estado de “*vagueando*” (2). Há uma analogia com o processo interno desencadeado pelo inseto, que, de forma inconsciente e incondicional, faz com que seus sensores olfativos comecem a ficar saturados. Atingindo-se a saturação máxima, o inseto perde ou se esquece de seu objetivo inicial, seguindo para o estado 2. O termo “*consciência*” (“self”) já foi empregado em [Mey86], com o objetivo de descrever um modelo interno de conhecimento sobre o mundo pictorial externo, na estrutura funcional da base de conhecimento de robôs móveis inteligentes. Recentemente, John McCharty explora o termo “*consciência*” para conceitos de estados mentais em robôs [McC95].

Uma máquina semelhante, que mapeia entradas sensoriais versus locais visitados, foi utilizada em [Kae95] na identificação de sistemas dinâmicos segundo uma seqüência de pares de entradas/saídas do autômato. A identificação de sistemas refere-se à inferência de um modelo dinâmico, que governe a interação entre o agente e o seu ambiente. A proposta do trabalho de K. Basye e outros [Kae95] é encontrar um conjunto de equações diferenciais ou um conjunto de regras de produção ou um conjunto de probabilidades de transição de estado de um processo estocástico. A aprendizagem do sistema de identificação é feita com uma máquina de Moore de estados discretos. Em [Kae95], os estados correspondem às entradas sensoriais, enquanto que os arcos ou saídas são as ações disparadas no deslocamento espacial. O método, porém, é limitado por um baixo número de entradas sensoriais e ações possíveis, o que evidencia as limitações de métodos matemáticos, que são restritos e não ap-

resentam boas perspectivas para o problema de AA's em ambientes desconhecidos [Ste94b, Ste94a, Ste94c].

Eventualmente, o AC para o protótipo Maja pode apresentar um baixo desempenho, uma das razões é a forma de como a caminhada aleatória se desenvolve. Basicamente, o passeio aleatório é determinante para que o processo cognitivo funcione bem. Ao final desta tese, há uma sugestão em torno dessa questão, propondo um passeio aleatório “ordenado”, visando um melhor desempenho. No AC em questão, necessita-se de uma caminhada que realmente seja randômica. Uma outra consideração nesse nível é com base no teorema clássico da caminhada aleatória [Ste88, Chu74]:

“Iniciando-se de um ponto qualquer em uma caminhada aleatória restrita em um espaço finito, nós podemos alcançar outro ponto qualquer um número de vezes”.

Aparentemente, esse teorema demonstra um lado ineficiente, tratando-se da navegação de robôs. Contudo, Luc Steels [Ste88] explora o lado potencial característico existente no teorema, tratando-se de vários robôs empenhados na ação de caminhar aleatoriamente e de contornar obstáculos. Luc Steels [Ste88] enaltece as vantagens da caminhada aleatória, como sendo flexível, robusta, cognitivamente econômica e explora um paralelismo quando se tem vários robôs simultâneos. No modelo cognitivo proposto ao protótipo Maja, a robustez e o baixo custo são pontos comuns às características apontadas por L. Steels.

5.8 A Arquitetura Completa

Nesta seção apresenta-se como os três níveis descritos – *reativo*, *instintivo* e *cognitivo* – se acoplam, seguindo a estrutura da figura 5.1. A sumarização dos níveis numa só figura é a arquitetura de base para os protótipos dos AA's aqui desenvolvidos. A arquitetura completa é detalhada na figura 5.14. O bloco da “Estratégia Cognitiva” é válido apenas para o protótipo *Maja*. Nesse bloco encontra-se o autômato comportamental descrito pela figura 5.13.

O bloco de “Habilidades Cognitivas” é válido para os três protótipos, definindo o nicho de atuação do AA. Em “Habilidades Cognitivas” encontra-se o arquivo-texto que contém a descrição dos agentes instintivos habilitados e seus respectivos limiares de disparos. O nível instintivo possui agentes que descrevem esquemas otimizadores de comportamento do AA. No nível reativo, encontra-se o circuito de agentes nebulosos definidos pela figura 5.7. As setas unidirecionais do nível instintivo para o reativo definem os esquemas a serem computados para um contexto, visando uma ação combinada.

O seletor diferencial (simbolizado por “▷” na figura) e o nebuloso (idem para “▷”) encontram-se no nível instintivo. O símbolo “⊗” representa o intercâmbio de dados existentes entre os dois níveis. A rigor, “⊗” opera como um limitador do deslocamento

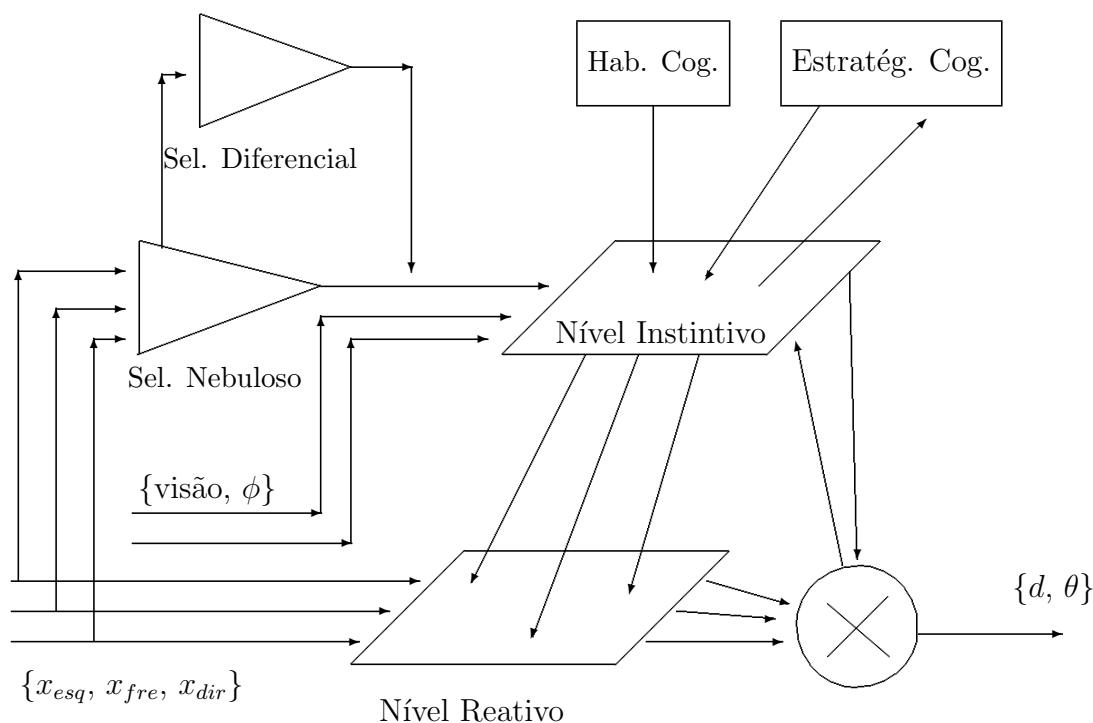


Figura 5.14: Arquitetura Completa

“ d ”. Eventualmente, a ordem de grandeza dos valores com que os agentes nebulosos foram implementados pode ser muito acima do limite de segurança estabelecido pela amplitude dos cones sensoriais. Isso evita que um valor inferido ultrapasse os critérios de segurança em um próximo passo do AA. O limite foi estabelecido em $0.5 \cdot Amp$, isto é, um deslocamento máximo de até 50% da amplitude dos cones. As definições dos agentes nebulosos encontram-se normalizadas com valores entre 0 e 100. Suas curvas de pertinência, bem como suas bases de conhecimentos, são apresentadas no apêndice.

Observa-se que o nível reativo processa apenas as informações obtidas pelos sensores de proximidade. Isso reforça a afirmação de que o nível reativo é responsável em processar dados pertinentes à vizinhança do AA. Os sensores de “visão” e “ ϕ ” são receptores especializados, e fornecem dados aos vários agentes do nível instintivo.

O seletor nebuloso tem prioridade devido a sua precisão em escolher um agente que reúna maiores perspectivas locais de uma boa ação. Caso nenhum dos agentes disponíveis no nível instintivo seja selecionado, o fluxo das entradas x_{esq} , x_{dir} e x_{fre} é utilizado pelo seletor diferencial. Suas características são a imprecisão (quase que “cego”) e a rapidez. Contudo, apesar de sua escolha quase que “ad-hoc”, os resultados experimentais são aceitáveis (capítulo 6).

5.8.1 Observações Finais

Finalmente, ressaltam-se algumas características imediatas, pertinentes à arquitetura proposta:

1. *Heterogeneidade*: devido à presença de agentes nebulosos, baseados em regras de produção representando esquemas internos e uma inspiração biológica na conexão entre eles;
 2. *Distribuída*: há vários agentes nessa arquitetura. Alguns combinados de forma cooperante e/ou competitivos, nos níveis reativo e instintivo. Não há presença de um agente ou um esquema centralizador para decisões globais;
 3. *Hierárquica*: os três níveis dessa arquitetura lhe conferem critérios de funcionalidades intrínsecas de projetos de AA's;
 4. *Instintiva*: devido ao extenso uso do conceito de esquemas [MAA95, Ark93], eles foram utilizados para definir estereótipos da natureza instintiva de um ser biológico;
 5. *Flexibilidade*: devido à facilidade de incorporar novos tipos de sensores e agentes à arquitetura;
 6. *Adaptação*: por poder usar essa arquitetura em outros domínios de problemas complexos;
 7. *Validação conceitual sobre arquiteturas em IAD*: os conceitos tratados nessa arquitetura são pertinentes à IAD.
-

Capítulo 6

Implementações e Resultados

6.1 Introdução

Neste capítulo são apresentados os resultados das simulações realizadas sobre os três protótipos: Brooks, Steels e Maja. O capítulo é dividido em duas partes: na primeira parte são realizadas algumas considerações comuns sobre as implementações dos três AA's. Os pontos aqui levantados tangenciam a formalização computacional utilizada na construção dos protótipos. Na segunda parte do capítulo, são apresentados os resultados relativos à locomoção dos AA's em diversos ambientes.

6.2 Considerações Iniciais

Nesta seção, alguns pontos são detalhados sobre a implementação computacional dos protótipos, seu esquema de coordenadas da navegação, deslocamentos e rotações do agente num espaço plano qualquer.

6.2.1 A Implementação

Os três protótipos foram construídos em ambiente Windows PC (Windows 3.1, 3.11 e 95) nas linguagens Visual Basic e C. O código em C foi responsável pelas máquinas de inferência nebulosa, pois os agentes reativos necessitavam de eficiência e velocidade em seus cálculos. O número de linhas de código por protótipo foi da ordem de 3.000 linhas. Esse número foi em função da quantidade de rotinas gráficas, que envolvem uma simulação nos níveis de detalhamento abordados nos protótipos. Por exemplo, o sensor de olfato ou de som foi esquematizado segundo uma reta que interliga a fonte atratora ao agente. De modo aproximado, esse valor poderia simplesmente ter sido passado com auxílio de uma tabela, o que não foi o caso. O protótipo Maja consumiu algumas centenas de linhas a mais, em virtude de seu nível cognitivo possuir um autômato comportamental. As interfaces são amigáveis, mas sem otimizações.

6.2.2 Coordenadas de Navegação

Praticamente todos sistemas móveis apresentam algum tipo de coordenadas referenciais. As marcas referenciais são importantes para que um próximo estado possa ser atingido ou que importantes estados anteriores possam ser recuperados. A exemplo das abelhas, que apresentam um esquema de memorização para regiões importantes em seu habitat [HW93]. A memorização nas abelhas é destinada ao reconhecimento de arredores sem a perda de um referencial. O esquema de navegação das abelhas para distâncias é o de integração do caminho (em inglês é comumente referenciado por “*dead reckoning*”), em que o sol é usado como referencial para variação angular e a distância é avaliada por um esquema sensorial interno. A comunicação de uma abelha sobre uma nova fonte de comida é feita via um ritual de giros em torno de si, acompanhado pela movimentação assíncrona de suas asas. Há um esquema de comunicação que fornece uma informação aproximada da localização da nova fonte de comida. Um estudo sobre esquemas de locomoção orientada por odores é encontrado em E. A. Arbas e outros [WK93]. O olfato é uma importante característica de orientação referencial, usado pela maioria dos animais, contudo, mais observável nos mamíferos [WK93].

O esquema de orientação com ângulos e distâncias é o utilizado nesta tese. Essas duas medidas são mínimas, mas de um bom poder de representação. O esquema aqui adotado segue aproximadamente a maioria dos trabalhos na área de robótica [HW93, BeM94, Zel88, SW95]. Tratando-se de alinhamentos de pontos, os métodos podem apresentar variações [FL94]. Contudo, a necessidade de um esquema referencial de orientação é pelo fato desta tese ter sido desenvolvida em ambiente simulado e não real. Os experimentos físicos sem processos de memorização ou sem modelagem do mundo real, a priori, não necessitam de nenhum esquema de orientação. Esses experimentos são fortemente guiados pelos dados de seus sensores, combinando uma ação efetora e tarefa a cumprir. Os exemplos típicos são os agentes reativos (ver capítulo de 3).

O esquema de localização é descrito a partir de um ponto inicial de partida para o AA. Na figura 6.1, um ponto inicial qualquer de partida no mapa é dado por (x, y) . Um próximo ponto é inferido a partir de uma ação do AA, dada pelo par (θ, d) da equação 5.4. Os valores angulares θ_i do AA são referenciados a partir da linha média do cone frontal com o eixo das abscissas (eixo “ x ”) do sistema cartesiano. A linha mediana do cone sensorial dianteiro é um eixo fixo, usado para orientar o AA em relação ao sistema cartesiano. Sendo assim, os valores para θ_i ’s são sempre absolutos, pois representam a diferença entre os dois eixos (das abscissas e da mediana do cone frontal). A distância d corresponde a um deslocamento positivo à frente da nova direção calculada.

Para um primeiro ponto, a distância é dada por d_1 e um ângulo θ_1 (ver figura 6.1). O ângulo θ_1 é um desvio angular, considerado a partir da linha mediana do cone frontal. Inferido o par (θ_1, d_1) , um novo ponto no espaço é obtido: (x_1, y_1) . A partir de (x_1, y_1) , um novo par (θ_2, d_2) é obtido como saída inferencial do agente. Essa variação e a distância levam a um novo ponto: (x_2, y_2) . O processo se repete

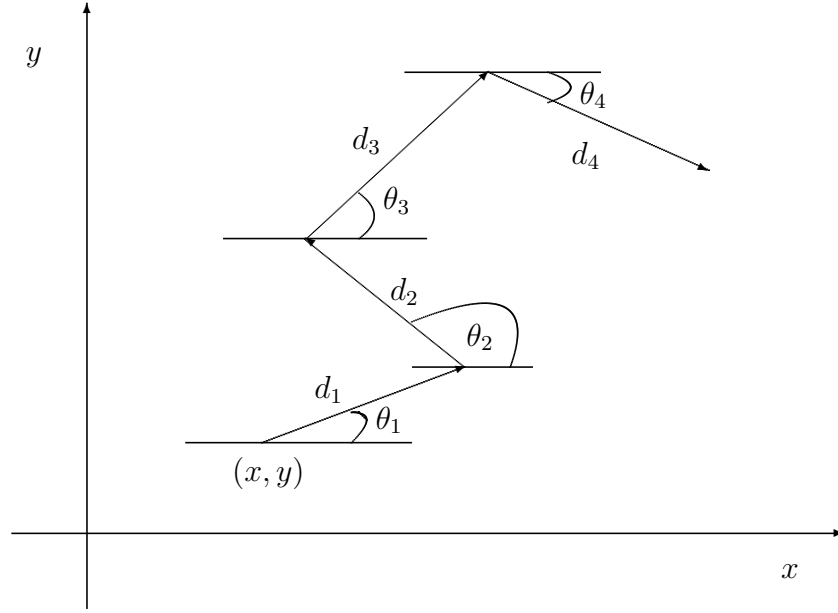


Figura 6.1: Mapa gerado por ângulos e distâncias

sucessivamente até o final da simulação (ver figura 6.1).

Em resumo, tem-se um esquema de cálculo de trajetória do AA, em função de dois parâmetros: uma distância (d_i) e uma variação angular (θ_i). Generalizando-se os passos para um ponto (x_{n-1}, y_{n-1}) , com valores d_n e θ_n , uma nova coordenada (x_n, y_n) é obtida pelas equações abaixo:

$$x_n = x_{n-1} + d_n \cos(\Delta\theta_n) \quad (6.1)$$

$$y_n = y_{n-1} + d_n \sin(\Delta\theta_n) \quad (6.2)$$

onde $\Delta\theta_n$ é a diferença angular corrente. O valor de $\Delta\theta_n$ sumariza todas as variações angulares do AA desde o início de sua trajetória, dado por:

$$\Delta\theta_n = \theta_n - \Delta\theta_{n-1} \quad (6.3)$$

Reforçando, $\Delta\theta_n$ expressa um resumo de todos os valores de θ_n desde o início da simulação, com a linha mediana do AA coincidindo com o eixo das abscissas no sistema cartesiano. Vale salientar que, no início, os dois eixos encontram-se em paralelo, implicando em $\Delta\theta_0 = 0$. O primeiro posicionamento do AA é feito com o uso do “mouse”, sobre qualquer ponto no ambiente simulado. Isto é, há um alinhamento inicial de sentido e direção entre os eixos das abscissas e da linha mediana do cone frontal. Para se realizar ajustes em sua posição inicial, há dois botões que controlam o AA, deslocando-o e rotacionando-o para situações particulares de partida. Em resumo, o detalhe semântico é que $\Delta\theta_n$ expressa um resumo histórico de

todas as rotações que o AA realizou no espaço cartesiano. O valor de $\Delta\theta_n$ é relativo e, conhecendo-se seu valor, sabe-se para onde o cone frontal do AA está direcionado.

Arelado ao direcionamento do AA, há o problema da convenção angular. A contabilização angular deve ser única. O giro angular deve ser convencionado, bem como foi a origem onde se utilizou o eixo das abscissas. A equação 6.3 apresenta uma convenção anti-horária para ângulos positivos e horária para ângulos negativos, estabelecida nesta tese. Para um sentido oposto, em que ângulos positivos seguem o sentido dos ponteiros do relógio, a equação 6.3 teria o seu sinal “-” trocado por “+”. Em suma, adotou-se o sentido anti-horário para os ângulos positivos, conforme a equação 6.3.

6.2.3 Amarração dos Pontos

De acordo com a definição física do AA, os três cones sensoriais são modelados por três triângulos, que possuem um vértice em comum. O vértice em comum representa o centro de gravidade do AA. O conjunto de pontos que formam os vértices dos triângulos são mostrados na figura 6.2.

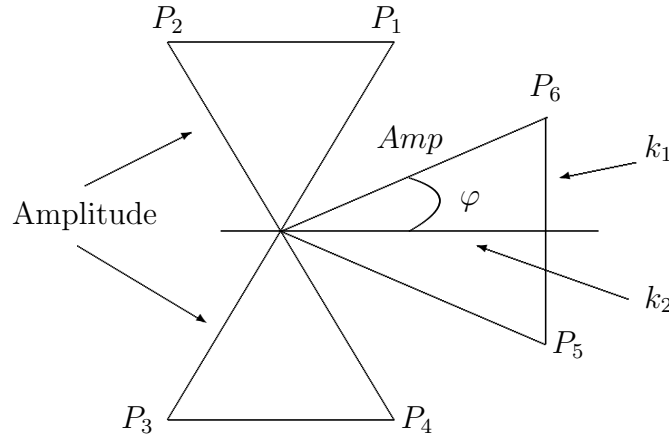


Figura 6.2: Vértices dos cones do AA

De acordo com a figura 6.2, os pontos P_1 e P_2 delimitam o feixe esquerdo; P_3 e P_4 , o feixe direito; P_5 e P_6 , o feixe frontal. Há o ponto P_7 que é a interseção dos três triângulos. A figura 6.2 pode ser representada por um conjunto de coordenadas, dadas sob a forma de matriz:

$$(P_1, P_2, P_3, P_4, P_5, P_6, P_7) = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \\ x_5 & y_5 \\ x_6 & y_6 \\ x_7 & y_7 \end{bmatrix} \quad (6.4)$$

onde $(x_7, y_7) = (x, y)$ é a coordenada do ponto central do AA.

Com relação a uma posição qualquer (x, y) do AA, as demais coordenadas são indexadas pelos catetos do triângulo-retângulo, formado a partir da metade de um cone sensorial, vista em detalhe na figura 6.3.

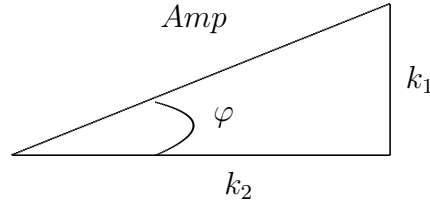


Figura 6.3: Detalhe da metade de um cone sensorial

Observar que o ângulo φ corresponde à metade da abertura completa de um feixe ou cone sensorial. Portanto, as coordenadas P_1 a P_6 que delimitam os cones sensoriais do AA são obtidas por uma relação fixa entre si, dadas por:

$$P_1 = (x_1, y_1) = ((x + k_1), (y + k_2)); \quad (6.5)$$

$$P_2 = (x_2, y_2) = ((x - k_1), (y + k_2)); \quad (6.6)$$

$$P_3 = (x_3, y_3) = ((x - k_1), (y - k_2)); \quad (6.7)$$

$$P_4 = (x_4, y_4) = ((x + k_1), (y - k_2)); \quad (6.8)$$

$$P_5 = (x_5, y_5) = ((x + k_2), (y - k_1)); \quad (6.9)$$

$$P_6 = (x_6, y_6) = ((x + k_2), (y + k_1)); \quad (6.10)$$

De acordo com a figura 6.3, os valores de k_1 e k_2 são obtidos por:

$$k_1 = Amp \cdot \sin \varphi \quad (6.11)$$

$$k_2 = Amp \cdot \cos \varphi \quad (6.12)$$

O valor “*Amp*” (*Amp* de *amplitude*) é o alcance da lateral de um dos cones sensoriais (ver figura 6.2). Por simplificação, os três cones possuem as mesmas medidas nessas laterais. Uma aproximação que na prática é bem factível, pois normalmente sensores de um mesmo tipo possuem as mesmas especificações técnicas. A nível de interface dos programas simuladores, os valores de “*Amp*” e φ são ajustados graficamente pelo usuário, em face das dimensões do ambiente em que se deseja processar uma simulação.

6.2.4 Cálculo da Rotação

A rotação é calculada sobre o agente quando se encontra num ponto qualquer (x_{n-1}, y_{n-1}) com uma diferença angular $\Delta\theta_{n-1}$ em relação à origem. Dada uma nova inferência ou ação, os novos valores d_n e θ_n são obtidos como saída do AA. O cálculo da nova coordenada (x_n, y_n) cartesiana do agente é fornecido pelas equações 6.1 e 6.2. Contudo, falta considerar a rotação dos pontos P_1 a P_6 em relação ao novo ponto (x_n, y_n) . Há necessidade de uma reorientação no esquema das coordenadas dos três triângulos em torno de P_7 ou (x_n, y_n) . A rotação angular necessária é dada por $\Delta\theta_n$, derivada a partir da equação 6.3.

O objetivo é calcular as novas coordenadas de um conjunto de pontos rotacionados por um ângulo α qualquer. Como já descrito, o conjunto de pontos expressa o modelo físico do AA (ver figura 6.2). A rotação do conjunto ocorre em relação a um ponto genérico (m, n) , que é o centro do AA. Na rotação de um conjunto de pontos em relação a um ponto fixo, utiliza-se do método da matriz de *coordenadas homogêneas* [Fil87], o que permite que novos pontos sejam encontrados após uma rotação por um ângulo α em relação a um ponto qualquer (m, n) no espaço cartesiano. Lançando-se mão dessa técnica no problema acima descrito, tem-se:

$$\alpha = \Delta\theta_n$$

$$(m, n) = (x_n, y_n)$$

Considerando que os 6 (seis) pontos que definem os 3 (três) triângulos conservam uma relação fixa entre si (ver equações 6.5 a 6.10), tem-se que $P_7 = (m, n)$, sendo uma “âncora” para todos os demais pontos. O reajuste ocorre segundo a matriz de rotação de *coordenadas homogêneas* [Fil87], dado por:

$$\begin{bmatrix} X & Y & H \end{bmatrix} = \begin{bmatrix} x_i & y_i & 1 \end{bmatrix} \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ n \sin \alpha - m(\cos \alpha - 1) & -m \sin \alpha - n(\cos \alpha - 1) & 1 \end{bmatrix} \quad (6.13)$$

O termo $\begin{bmatrix} x_i & y_i & 1 \end{bmatrix}$ da equação 6.13 é válido para $i = \{1, 2, 3, 4, 5, 6\}$, isto é, as coordenadas de P_1 a P_6 , definidas segundo as equações 6.5 a 6.10. Após a rotação, as novas coordenadas são dadas pela matriz $[X, Y]$. A implementação computacional

foi feita com um laço repetitivo para cada coordenada (x_i, y_i) , isto é, individualmente para os valores das equações 6.5 a 6.10. O valor de $H = 1$ satisfaz à rotação planar. Caso $(m, n) = (0, 0)$, a matriz acima é reduzida a uma rotação com referencial sobre a origem $(0, 0)$.

6.2.5 Cálculo da Interseção dos Cones com Objetos

Tratando-se de uma navegação com orientação local, sem referências espaciais de objetos ou descrição do ambiente, procedimentos são necessários junto aos sensores do AA para que ele se situe em uma região do ambiente. A todo instante, o agente pode estar afastado ou próximo de paredes e/ou de objetos ou, eventualmente, chocar-se com as paredes.

Os cones sensoriais fornecem alguns dados ao agente sobre sua vizinhança. O agente tem apenas uma noção quanto às suas redondezas. Os sensores não procuram traduzir ou interpretar os obstáculos para uma representação interna de modelo. Eles são dados numéricos com uma semântica de proximidades. Há um acoplamento direto entre um número fornecido diretamente pelos sensores e uma ação motora compatível, selecionada a partir dos agentes instintivos. Para se efetuar tal detecção, métodos velozes são necessários, uma vez que os protótipos são simulados.

Para isso, implementaram-se procedimentos que detectam objetos de modo eficiente e a baixo custo. O formato geométrico do agente, delimitado pelo conjunto de pontos $[P_1, P_2, P_3, P_4, P_5, P_6, P_7]$, permitiu construir um conjunto de equações de retas. O conjunto de equações de reta formam as partes físicas do AA junto ao simulador, as quais definem os três triângulos sensoriais.

Desejando-se uma navegação de baixo custo computacional, factível em um computador pessoal, optou-se em não avaliar a área formada pelos triângulos. O cálculo real da interseção da área entre os triângulos e objetos tornaria inviável a construção de um protótipo em tempo real, pois analisar duas áreas sobrepostas é um problema de alto custo computacional, envolvendo áreas como processamento de imagem e/ou análise de figuras ou contornos. Contornando o problema de custo e complexidade computacional, decidiu-se apenas avaliar o perímetro desses triângulos quando sobrepostos a objetos. Isso porque as equações de reta dos triângulos se encontram disponíveis a partir dos pontos das equações 6.5 a 6.10. Para cada cone estimou-se uma avaliação de 33 pontos, 11 para cada reta dos triângulos. Tal valor foi estimado com objetivo de operar em qualquer tamanho de monitor de vídeo. Eventualmente para grandes ambientes, deseja-se um agente um pouco maior ou vice-versa. Alguns testes foram feitos em torno da escolha, 11 pontos por reta, e os resultados foram praticamente idênticos, tanto para números maiores como menores de pontos sobre a reta. A distância entre os pontos ao longo da reta encontram-se numa distribuição uniforme. O espaçamento ou incremento entre os pontos é calculado em 10% do comprimento da reta corrente. Obviamente houve uma redundância de cálculos sobre alguns pontos nos cones, por exemplo, sobre o ponto central. Contudo, tal redundância é justificada pela simplificação computacional nos programas.

6.3 Protótipo Brooks

Na figura 6.4, Brooks é submetido à navegação próximo a objetos irregulares, onde o cone sensorial da direita encontra-se habilitado. Tal fato implica em afirmar que os agentes instintivos do lado esquerdo do agente estão desligados ou inativos.

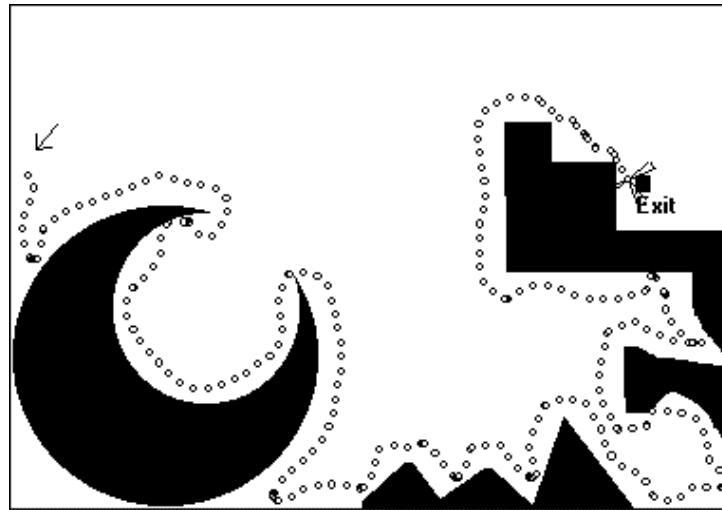


Figura 6.4: Navegação em ambiente “lunar”

Apesar das interseções dos cones serem calculadas com base no seu perímetro, isto é, uma medida aproximada, observa-se que o agente demonstrou robustez ao contornar os cantos da “meia-lua” na figura 6.4. O comportamento do agente na figura 6.4 é praticamente idêntico a um dos resultados obtido em [AD91]. O agente móvel proposto por T. L. Anderson e M. Donath [AD91] apresenta uma variante das arquiteturas de englobamento [Bro91c, Bro91d], onde os comportamentos são combinados segundo uma natureza espacial (objetos e ambiente) e um ordenamento temporal. Observa-se que o trecho final da trajetória é quase uma reta. Isso deve-se ao fato de a cada instante o agente procurar pela “porta de saída”. Uma das entradas no nível instintivo é a possibilidade de enxergar ou não uma marca. Na figura 6.4, ela é representada por um quadrado pequeno. A “visão” simulada foi implementada utilizando-se de uma reta entre o agente e a marca (porta) de saída. A todo instante calcula-se uma interpolação de pontos sobre a reta para detectar a presença ou não de obstáculos. A seta na figura 6.4 indica aproximadamente o ponto de partida do agente. Essa seta como ponto de partida é válida nas figuras seguintes.

Na figura 6.5, o agente percorre um labirinto típico. Nesse exemplo habilitou-se os agentes instintivos do lado esquerdo. Ou seja, nesse caso a navegação próxima às paredes é realizada com o cone esquerdo sempre em contacto. Alguns pontos da trajetória na figura 6.5 tocam as paredes. Esse fato, propositalmente aqui apresentado, ocorre por alguns motivos:

- Devido ao tamanho dos cones do agente versus o ambiente: se o agente for muito grande em relação a um corredor pode ocorrer a perda do referencial;

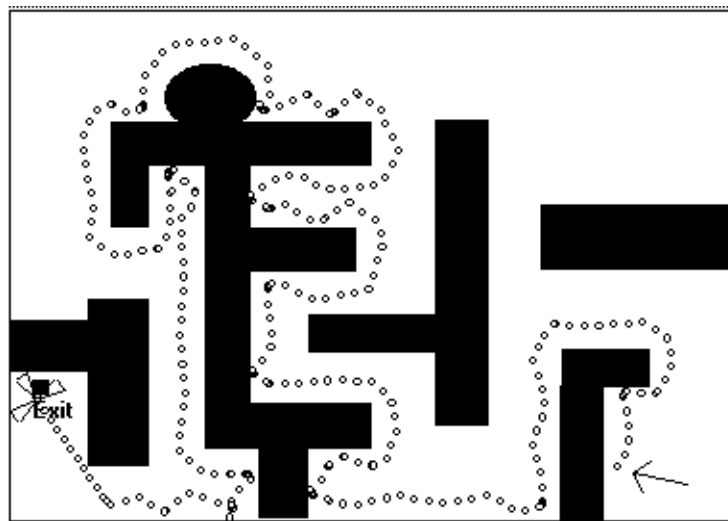


Figura 6.5: Ambiente típico de um labirinto

- Devido aos limiares de disparo definidos aos agentes instintivos no nível cognitivo; tais limiares são aproximados e não foram otimizados.

Mais recentemente, uma navegação rudimentar em labirintos “regulares” tem sido estudada por R. Sun e outros em [Pet95, Mer96, PM96]. Contudo, o objetivo é a aprendizagem por reforço sobre seqüências de episódios a fim de obter regras de produção para uma arquitetura híbrida. Os labirintos estudados nestes trabalhos [Pet95, Mer96, PM96] são mais simples que o apresentado na figura 6.5. A aprendizagem por reforço a partir de seqüências simples de episódios visa melhorar gradativamente as regras de produção em nível cognitivo, que vão ser aplicadas em labirintos mais complexos. Complexos no sentido de serem maiores, mas simples na sua formação estrutural. Os labirintos de R. Sun são compostos por um corredor principal e vários secundários do tipo beco (sem saída). O agente de R. Sun e outros também possuem três sensores, um em cada lateral e um na parte frontal. Contudo, esses sensores estão habilitados a identificar três tipos de dados: um corredor, uma parede e uma saída (uma marca de fim). Os movimentos permitidos pelo agente são: girar à esquerda, girar à direita e seguir em frente. Os objetos a serem identificados são simples (determinados num número total de três), pois o objetivo é realizar várias seqüências que possam levar a uma recompensa quando o episódio for bem sucedido e uma punição em caso contrário. O desempenho quanto às trajetórias obtidas com a aprendizagem “bottom-up” não são apresentadas, pois o principal objetivo é tratar (derivar, generalizar e revisar) regras de produção a nível de regras desta arquitetura. Oportunamente, no capítulo 7, discute-se a questão da aprendizagem em AA’s.

6.4 Protótipo Steels

Por se tratar do protótipo mais interessante quanto a sua funcionalidade emergente e de aplicações práticas, um maior número de exemplos são apresentados. Nesse caso há um maior número de agentes instintivos habilitados a executarem suas ações. Ao contrário do protótipo anterior, aqui os agentes habilitam esquemas à esquerda, à direita e à frente, que competem em igualdade de condições por uma ação. Como já frisado, essa ação tem origem no acionamento de um esquema motor sobre o nível reativo, a partir da seleção de um dos agentes do nível instintivo, selecionados de acordo com o conjunto das entradas sensoriais.

Na figura 6.6 observam-se as quadras ou blocos (os quadrados maiores) que formam um centro da cidade. Ao se caminhar em ambientes com essas características, normalmente toma-se um ponto de referência, por exemplo, o luminoso do prédio mais elevado. A partir dessa referência procura-se caminhar em direção aos alvos ou marcas (representado pelos quadrados menores). Nos trechos em que trajetória segue uma linha reta, ela representa uma ausência de perturbação aos sensores laterais pela presença de obstáculos, em relação à direção a seguir. Na figura 6.6 foram definidas três marcas para o agente encontrar.

No trabalho de Y. Zhao e outros [BeM94] há um ambiente típico de centro de cidade como o aqui apresentado, exceto que o ambiente possui vários robôs caminhando em direção a um alvo comum, mas comunicando-se com uma base comum de informação, que os provêem de dados sobre os caminhos a seguir. Há uma exploração distribuída e quase descentralizada de um território via uma central de trocas de informações. Contudo, o objetivo de Y. Zhao e outros [BeM94] é investigar o tamanho das memórias principal e “*cache*”, usando a memória auxiliar e banda-passante na comunicação entre os robôs e a central.

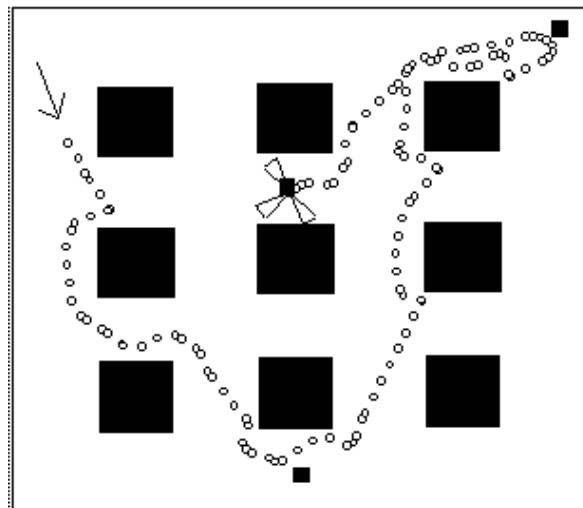


Figura 6.6: Ambiente típico do centro da cidade

A figura 6.7 lembra a bandeira brasileira ou uma mesa circular e quatro bancos.

A proposta da figura 6.7 foi esconder as marcas de modo que o agente tivesse que contornar as diversas figuras geométricas que a compõem. Novamente, nos trechos em que há uma linha reta, o agente buscou minimizar a diferença angular entre a marca e a linha mediana de seu sensor frontal. Experimentos que visam contornar objetos segundo o gradiente potencial das áreas livres são usados na área de AAs [Ark91, Ark93, Pay91]. Dentro dessa linha de uma navegação imprecisa com campos potenciais dos sonares, encontra-se o trabalho de [FL94]. A figura 6.7 foi derivada a partir dos mapas esquemáticos de áreas navegáveis versus obstáculos apresentados em [FL94].

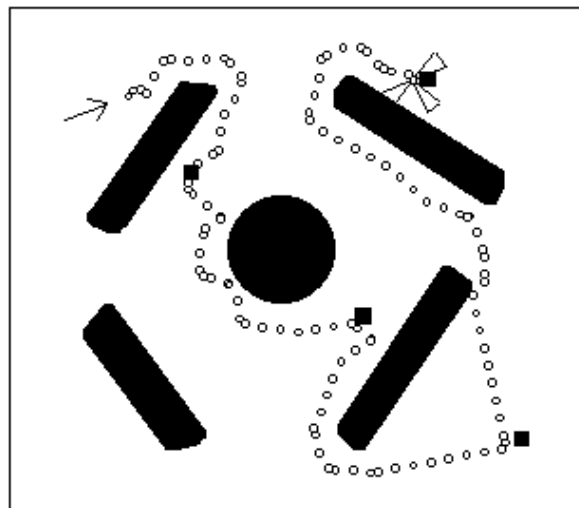


Figura 6.7: “Bandeira do Brasil”

Na figura 6.8 é apresentado um conjunto de prateleiras que lembram a formação do ambiente interno de um supermercado. As marcas (os quadrados menores) são distribuídas em pontos estratégicos, como esquinas de prateleiras, que dirigem o agente a percorrer os corredores do supermercado numa determinada ordem. A figura 6.8 fornece uma idéia da proposta prática que esse protótipo apresenta.

O ambiente apresentado nas figuras 6.9 e 6.10 foi retirado de [GR95]. A diferença das figuras 6.9 e 6.10 é a posição da marca final em relação à posição inicial do agente. Na figura 6.9, a marca final encontra-se abaixo da posição inicial, enquanto que na figura 6.10, a marca final encontra-se numa linha acima do ponto de partida.

Em [GR95], o problema do móvel autônomo é enfocado sob uma visão neuro-nebulosa. O controlador neuro-nebuloso proposto em [GR95] considera o ângulo frontal formado com um obstáculo mais próximo, sua distância e o ângulo do agente com relação à marca final. Três redes neurais são treinadas considerando o comportamento de evitar obstáculo, de seguir em direção à marca e de coordenar os dois comportamentos visando atingir uma marca ou meta. As redes neuronais formam grupos de repertórios individualizados que são encadeados segundo critérios de seleção, ou seja, a *teoria de seleção de grupos neuronais* [GR95] suporta uma arquitetura que apresenta aprendizagem em seus esquemas neurais. Para determinar

os parâmetros das redes neurais, utilizaram-se técnicas de algoritmos genéticos. Normalmente, os parâmetros definem o número de camadas, número de neurônios, bem como os pesos das conexões entre os mesmos. Os resultados obtidos em [GR95] não diferem significativamente dos resultados das figuras 6.9 e 6.10. Contudo, o custo computacional é muito maior que o protótipo aqui apresentado.

No último obstáculo da figura 6.10, o agente realizou algumas “*voltas sobre si mesmo*” (um “*loop*”), igualmente, em alguns cantos das figuras 6.9 e 6.10. Essas “*voltas sobre si mesmo*” indicam a limitação desse agente em ambientes que possuam objetos côncavos. Vale lembrar que o referencial do agente é a maximização na direção de uma fonte atratora. O último obstáculo da figura 6.10, leva a deduzir que o agente encontra-se em vias de entrar num estado colapso ou indefinido de “*voltas sobre si mesmo*”, pois optou em desistir do contorno na parte superior do obstáculo. Considerando que não há uso de uma função heurística como distância entre o agente e a fonte, nem mecanismos de memorização, conclui-se que ambientes como os das figuras 6.9 e 6.10 representam algum dos limites desse protótipo. Finalmente, vale observar que no último obstáculo da figura 6.10, em que o AA realiza uma exploração para ambos os lados do obstáculo, o comportamento obtido é típico ao da formiga quando perde o feromônio do rastro de um caminho. Impedindo-se fisicamente o caminho das formigas, elas realizam explorações laterais de forma independente e distribuída, até que a primeira formiga a ter sucesso de um caminho alternativo comunica às demais sobre o novo rastro a seguir. A comunicação entre duas ou mais formigas é realizada via uma sucessão de toques de suas antenas.

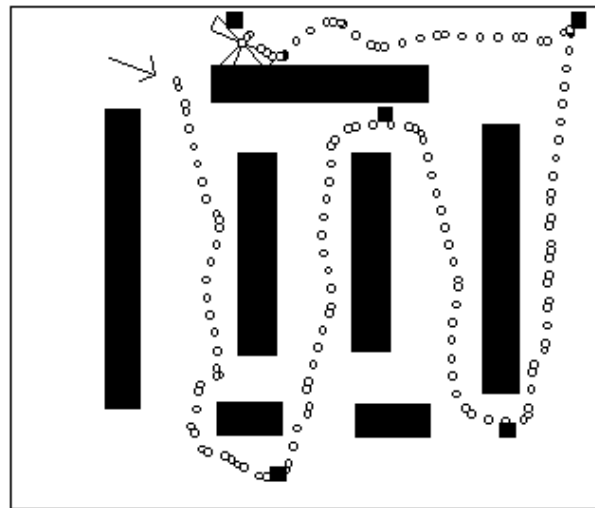


Figura 6.8: Ambiente supermercado

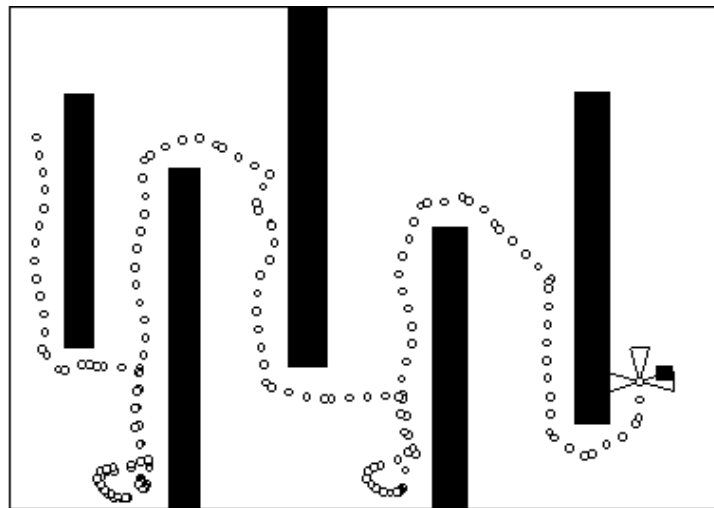


Figura 6.9: Ambiente com obstáculos frontais

6.5 Protótipo Maja

O desenvolvimento do protótipo Maja foi motivado pela limitação do protótipo Steels em contornar objetos côncavos (salas, becos, etc) em um ambiente qualquer. Quando um agente se encontra num desses locais, há necessidade de uma heurística extra de autolocalização. Essas heurísticas variam desde o uso da maximização de sensores radiais até mapeamentos de uma região, usando-se redes neurais [vP94c, Zim94, vP94a]. No capítulo anterior, esse limite é contornado com a introdução de um *autômato comportamental*, localizado no nível cognitivo do agente e visa alterar um tipo de comportamento intermitente. A técnica é de baixo custo computacional, mas adaptou-se

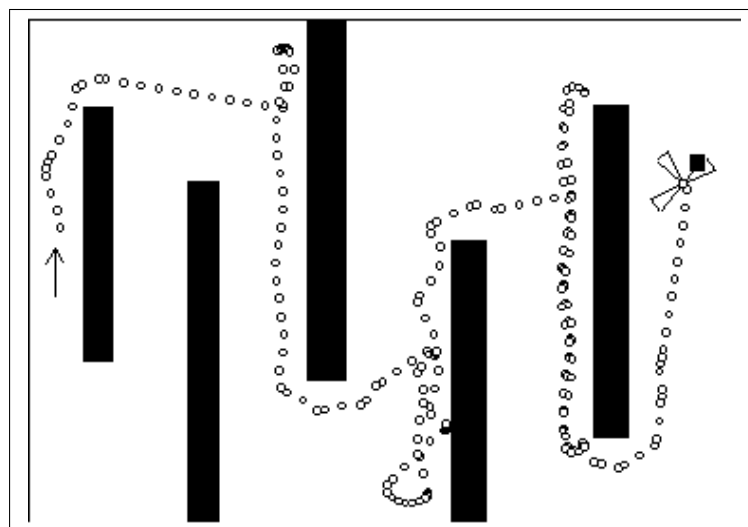


Figura 6.10: Ambiente com obstáculos frontais

facilmente à arquitetura aqui proposta.

Um objeto côncavo típico é apresentado na parte central da figura 6.11 (um “C” invertido). Caso o protótipo Steels entrasse em um beco como da figura 6.11 e sua fonte atratora estivesse no lado oposto à saída, simplesmente o agente estaria impossibilitado de encontrar a marca objetivada. Esse agente não apresenta competência de “*perceber*” que está caminhando em círculos numa dada região. A exceção é quando ele se encontrar na parte externa do objeto côncavo. Nesse caso, ele tem a competência de contornar qualquer objeto do tipo convexo¹, encontrando assim uma porta de entrada.

A trajetória inicial em linha reta da figura 6.11 descreve um direcionamento, maximizado do agente com o alvo, um comportamento típico de Steels. Ao encontrar-se no interior da sala, o agente visa caminhar em direção à fonte olfativa, contudo, de modo ineficaz, uma vez que existem três paredes limitando-o. O problema é contornado com o protótipo Maja onde há uma percepção de que ele se encontra caminhando em círculos. O agente Maja dispara então um novo comportamento, que é o de caminhar aleatoriamente ou vaguear². No capítulo anterior, o agente se automonitora com relação a um possível “*loop*” em torno de si numa dada região, permanecendo no estado vaguear por um número limitado de passos.

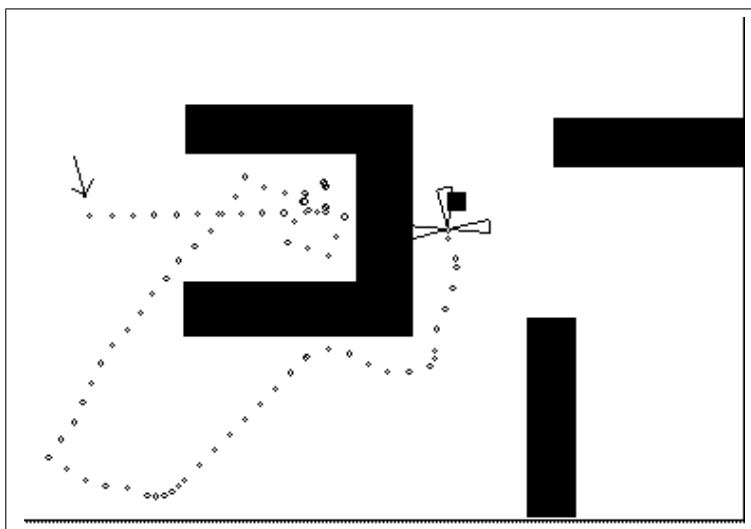


Figura 6.11: Ambiente de um beco com uma fonte atratora

A figura 6.12 apresenta um ambiente idêntico ao da figura 6.11, exceto que o número de marcas a serem atingidas sobe para três. Após a primeira marca, atrás do “C” invertido, o comportamento do agente é idêntico ao do protótipo Steels. Observa-se que o tempo de permanência no “C” invertido (número de pontos na trajetória) é maior que o da figura 6.11. A explicação é a aleatoriedade imposta pelo estado de

¹Isto é, a percepção é convexa vista pelo lado externo. Contudo, o agente desconhece o significado de paredes côncavas ou convexas.

²Ver figura 5.13.

vaguear, descrito no capítulo anterior. Uma nova simulação implicaria em um novo resultado em relação a figura 6.12. As figuras 6.11 e 6.12 são consideradas do tipo “*armadilhas*”, segundo A. Meystel [Mey86]. O agente de A. Meystel [Mey86] é baseado em regras de conhecimento, possuindo uma arquitetura próxima a conceitos da IA clássica e de robôs móveis, o qual conseguiria encontrar uma saída do “C” invertido se ele não apresentasse uma grande profundidade.

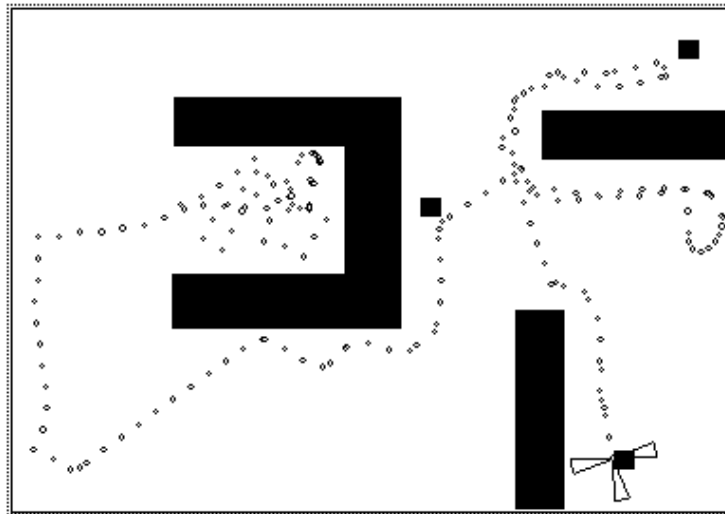


Figura 6.12: O agente visita outras marcas

A figura 6.13 foi reproduzida do trabalho de M. Oliveira e outros [GR95]. Em [GR95], essa figura, com várias salas, impõe um limite a seu agente quanto a sua arquitetura de controle baseada em redes neuro-nebulosas. Simplesmente, a aprendizagem não é completa para todos os tipos de ambiente com estratégias de navegação limitadas pelos dois comportamentos descritos (evitar obstáculos e seguir em direção ao alvo). Em [GR95], o agente permanece em “*loop*” em uma das salas da figura 6.13, pois o aprendizado dos comportamentos de seguir em direção à meta (à marca) e evitar obstáculos ficam restritos a ambientes semifechados. O agente necessitaria de um nível extra que monitorasse o seu estado de insucesso e, a partir desse fato, tirasse alguma vantagem, ou seja, um melhoramento em sua estratégia de navegação.

Na figura 6.13, o protótipo Maja obtém sucesso saindo de uma situação bem desfavorável: ele parte do lado oposto em que a meta se localiza. O estado de vaguear é propositalmente modelado com uma primeira opção de caminhar em frente com uma pequena aleatoriedade angular, de -20° a $+20^\circ$, seguido por uma segunda escolha à esquerda se possível. A escolha à esquerda segue de perto as definições dos esquemas, que visam acompanhar a parede esquerda, exceto se o sensor frontal detectar caminho livre, ele continua a caminhar em frente. Caso exista a impossibilidade de tomar à esquerda, um comportamento parecido ao esquema de seguir próximo à parede direita é disparado. Em resumo, para o estado vaguear, há uma prioridade do sensor frontal buscar movimentos à frente quando espaços livres forem detectados.

Na figura 6.14 e nos resultados seguintes, o protótipo foi alterado com objetivo de tornar randômico a escolha de tomar à esquerda ou à direita, quando o movimento de seguir à frente for inviabilizado. Uma chave comutadora aleatória foi incorporada com o objetivo de alternar uma escolha à esquerda ou à direita, sem conhecimento de suas escolhas passadas. No comportamento de vaguear, é desejável um comportamento aleatório em todas as direções de modo uniforme. Contudo, esse comportamento é difícil de ser obtido com um pequeno conjunto de simples regras de produção. Nesse caso, foram escolhidas regras que consigam, tanto quanto possível, manter o agente caminhando com uma certa aleatoriedade. Para isso foram desenvolvidas 4 (quatro) regras de produção, parcialmente priorizadas para controlar o estado de vaguear. A prioridade foi dada ao movimento à frente, seguido por uma escolha aleatória à esquerda ou à direita e, finalmente, a resolução de conflitos em paredes frontais ao agente. Em caso de uma parede frontal ao agente, há uma rotação aleatória de $+170^\circ$ a $+190^\circ$, seguida por um deslocamento de 10% da amplitude do sonar. Essa é a última regra a ser avaliada, quando os movimentos à frente, à esquerda e à direita forem bloqueados.

Nas figuras 6.15 e 6.16, o tamanho dos sonares foram diminuídos e o ângulo de abertura dos sonares foi aumentado em face da largura da porta nas salas abaixo. A sala das figuras 6.15 e 6.16 foi propositalmente construída na forma de um corredor com o objetivo de dificultar a passagem do agente. Melhores resultados seriam encontrados caso a porta estivesse em uma das laterais dessa sala.

A diferença de desempenho das figuras 6.15 e 6.16 encontra-se no tempo que o agente passa no estado de vaguear. A simulação da figura 6.16 foi subsequente à figura 6.15 e o temporizador do estado de vaguear não foi reiniciado. O temporizador é uma variável estática do programa, cujo objetivo é reproduzir processos Markovianos onde o estado corrente independe da história passada. Cada vez que o estado de vaguear é alcançado, o temporizador considera um número de passos ainda maior que a visita

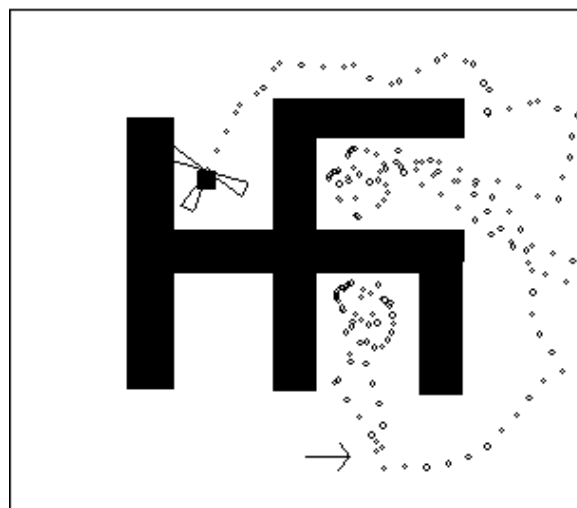


Figura 6.13: Saída tendenciosa pela esquerda

anterior a esse estado. O número inicial de passos é uma variável aleatória com um valor entre 5 e 10. Caso ocorra uma nova visita ao estado vaguear, o número inicial é adicionado por um novo valor aleatório entre 5 e 10, seguindo-se assim sucessivamente. Sumarizando tem-se:

$$t_n = t_{n-1} + rand(5, 10) \quad (6.14)$$

A escolha dos valores em 6.14 foi considerando uma média de 7 novos passos no estado vaguear. O valor sete (7) foi considerado para um caso de inexistência de obstáculos à frente, permitindo assim que o agente se desloque numa distância aproximada do triplo de sua amplitude visual máxima. Nesse estado, cada passo bem-

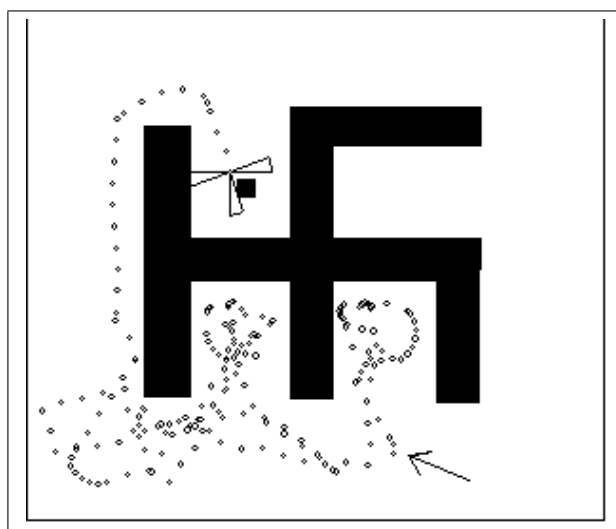


Figura 6.14: Aleatoriedade na escolha esquerda ou direita

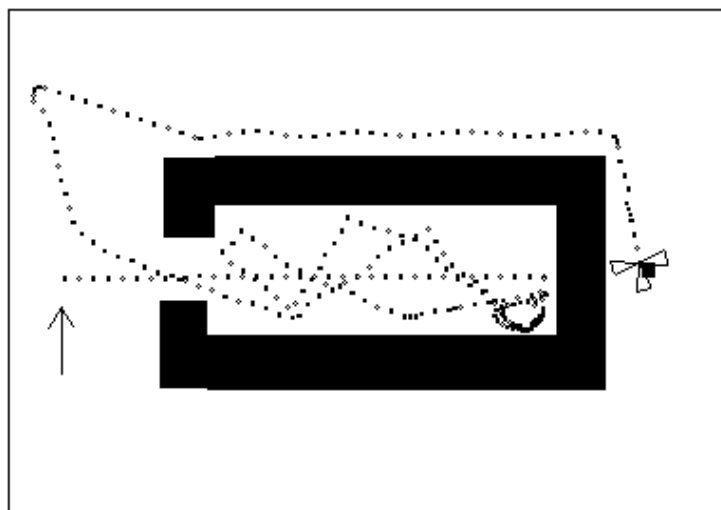


Figura 6.15: Sala quase fechada

sucedido à frente corresponde a 50% do comprimento da amplitude visual máxima.

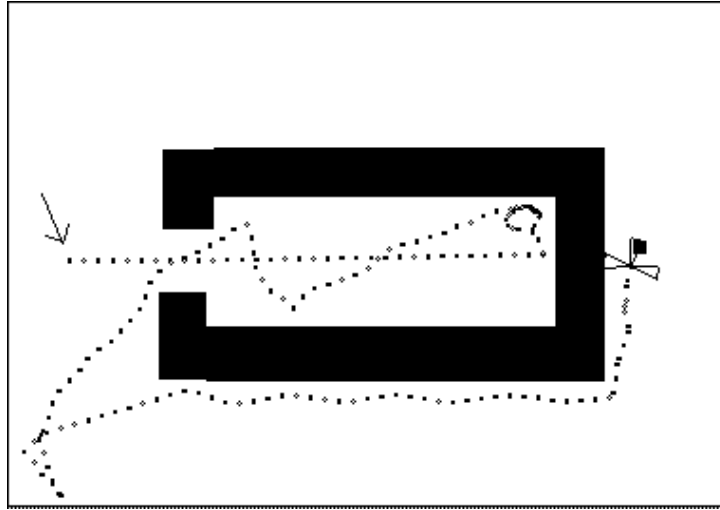


Figura 6.16: Tempo maior no estado de caminhar aleatoriamente

Capítulo 7

Conclusões

7.1 Introdução

Este capítulo possui outras quatro seções. Na segunda seção, há um resumo da arquitetura proposta com suas motivações e protótipos desenvolvidos. Finaliza-se a seção pela descrição dos níveis desta arquitetura. Na terceira seção, destacam-se os principais pontos alcançados com este trabalho e suas contribuições. Na quarta seção, há um comparativo entre alguns trabalhos correlatos e a arquitetura proposta, de maneira a complementar o que foi apresentado na segunda seção. E, finalmente, na última seção, delineam-se sugestões de futuras pesquisas na área de AA's.

7.2 Retrospectiva da Tese

O trabalho desenvolvido nesta tese propõe uma arquitetura para um agente autônomo (AA) do tipo móvel. A análise da mobilidade é escolhida em face ao grande número de aplicações práticas. Por outro lado, a locomoção é um dos comportamentos mais elementares dos seres vivos, mas que revela muitos aspectos de sua inteligência [RM93]. Os princípios aqui abordados estão relacionados a IA, IAD e AA's, mas imediatamente reusáveis em outros domínios de aplicações. Desse modo, a locomoção em seus vários aspectos apresenta-se como uma estratégia de estudo de caso.

Objetivando critérios de validação, três aplicações distintas foram instanciadas a partir da arquitetura proposta. A instanciação experimental dessas aplicações é uma estratégia metodológica de validação. A metodologia empregada foi a construção de protótipos sob forma de simuladores. Uma vez estabelecidos os requisitos de projeto, os resultados obtidos a partir das simulações tornam a arquitetura validada.

As três aplicações definem AA's sob uma mesma plataforma arquitetural, desempenhando funcionalidades dirigidas a nichos [HR95]. Isto é, há restrições quanto ao objetivo da aplicação e ao ambiente exploratório. Os três protótipos de AA's receberam nomes de *Brooks*, *Steels* e *Maja*, homenageando os pesquisadores da área. As funcionalidades desses protótipos são sucintamente descritas a seguir:

- A funcionalidade de Brooks é caminhar junto a uma parede, o que permite que saídas em labirintos típicos sejam encontradas. O AA possui a parede como seu único referencial, de modo que uma das restrições é a manutenção de permanente contato com a parede. Há um controle adaptativo quanto à locomoção e à presença de paredes com formatos desconhecidos;
- A funcionalidade de Steels é seguir uma lista de marcas em um ambiente desconhecido. As marcas definem as seqüências de metas a serem alcançadas (uma central de reabastecimento ou, em termos biológicos, uma fonte de comida, um parceiro para o acasalamento, etc). Essas marcas são espalhadas no ambiente e, a partir de um ponto qualquer, Steels procura-as na ordem em que foram especificadas;
- A funcionalidade de Maja é similar à de Steels, exceto que a habilidade cognitiva é mais evidenciada com o objetivo de detectar ciclos causados por becos ou topologias equivalentes. O agente Steels apresenta algumas restrições quanto aos ambientes explorados. Essas restrições referem-se à existência de objetos côncavos entre o agente e a fonte de motivação. Steels caminha bem em ambientes fechados como uma sala, um laboratório, supermercados, etc. Já o agente Maja torna-se genérico ao navegar em qualquer ambiente desconhecido. Para isso, ampliaram-se as suas habilidades cognitivas, mas sem considerar esquemas de armazenamentos de dados.

A arquitetura apresentada é hierarquizada em três níveis. A decomposição em níveis verticais visa uma melhor compreensão e detalhamento das funções internas do AA. De modo análogo, as arquiteturas comportamentais de AA's também apresentam a idéia de hierarquia em níveis e são fixas quanto às estruturas internas [Mat94]. Numa abordagem clássica da IA há uma decomposição horizontal nos procedimentos de percepção, modelagem, planejamento, execução da tarefa e controle motor [Bro86, MB90].

O primeiro nível é chamado de *reativo*. Esse nível consiste em uma rede de agentes nebulosos com uma topologia fixa, que exhibe um caráter intrínseco de ser adaptativa, competitiva, cooperativa e conflitante segundo uma visão da IAD. Nesse nível há um controle adaptativo do AA segundo as três entradas dos cones sensoriais de proximidade. Dois desses cones estão lateralmente direcionados e o terceiro possui uma visão frontal.

O segundo nível dessa arquitetura é chamado de *instintivo*. Esse nível é definido por um conjunto de agentes instintivos, distribuídos, sem nenhuma conexão entre si e atuam segundo um contexto ambiental oportunista. Os agentes instintivos foram construídos sob o paradigma de regras de produção, cujo conhecimento embutido é um esquema de controle motor sobre o nível reativo. Os esquemas modelam aproximadamente contextos ambientais, que obviamente são infinitos. Portanto, há necessidade de selecionar um agente instintivo que possua o esquema mais ajustado ao contexto, que leve a uma ação global coerente. O agente instintivo é selecionado a partir de

um classificador nebuloso, que considera as três entradas sensoriais. Adicionalmente, outras duas entradas sensoriais são colocadas à disposição dos agentes, consultadas quando houver interesse. Uma dessas entradas sensoriais é a visão direta entre o AA e uma marca, com valores *presente* ou *ausente*. A outra é o sentido de direção entre o AA e a marca, que simula uma importante habilidade de seres biológicos¹: “*a noção de direção*” de fontes ou atratores, como luz, alimento, sons, odores, etc.

O terceiro e último nível é o *cognitivo*. Nesse nível são definidos os agentes do nível instintivo devem estar ativos, visando uma cooperação coordenada e uma funcionalidade global. Os agentes cooperam e competem no nível instintivo, conduzindo o AA a uma funcionalidade intrínseca. Aqui permanece a idéia de que um macro-comportamento surge da combinação de vários e independentes micro-comportamentos [Bro93]. Esse nível foi modelado utilizando-se um arquivo-texto, no qual o usuário seleciona os agentes instintivos que devem estar presentes, visando algumas características comportamentais.

No protótipo Maja, um autômato comportamental foi explicitamente desenvolvido com o objetivo de detectar de regiões sem saídas para o AA. Esse autômato comportamental consiste em três importantes estados. O primeiro estado descreve o comportamento do AA em alcançar uma determinada marca. Contudo, há um contador de voltas em torno de si, que atua em paralelo. Quando esse contador atinge o limite de uma volta, contabilizado a partir de diferenças angulares, torna-se ativo um novo estado, o qual conta o número de voltas em torno de si. Há uma introspecção por parte do AA, em prestar atenção sobre seu comportamento a partir da primeira volta detectada. O estado é um comportamento típico de uma “*consciência*” rudimentar que o AA atinge [McC95, Slo96b, Slo96a]. Caso um número limite de voltas seja atingido, deduz-se que o AA está rodando em círculos numa região em particular do ambiente, seguindo, então, para um novo estado de navegação aleatória. Esse estado é uma locomoção sem destino ou a esmo por um certo período de tempo.

A movimentação itinerante pode seguir algumas heurísticas. A estratégia adotada pelo AA é um comportamento que evite permanências junto à parede, isto é, buscar movimentos em espaços vazios nas proximidades. Essa heurística foi escolhida em face do baixo custo computacional e sua construção foi feita a partir de regras muito simples. O comportamento aleatório de um AA é um problema aparentemente simples, mas com profundas implicações no contexto de agentes reativos.

Os autômatos comportamentais dos protótipos Brooks e Steels não foram explicitamente construídos. Esses autômatos foram inferidos a partir dos comportamentos obtidos da simulação. Um estado comportamental é uma regularidade contínua durante um período de tempo. A caracterização do estado de um AA formado por uma comunidade de outros agentes, a princípio, apresenta algumas dificuldades conceituais. Essas dificuldades são discutidas ao longo do texto, visando levantar estados comportamentais a “*longo prazo*” para os dois protótipos. Contudo, os três protótipos exibiram comportamentos regulares a “*longo prazo*”, permitindo a construção de autômatos descritores de sínteses comportamentais. A síntese resume a

¹Uma faceta mais evidenciada nos mamíferos.

seqüência de um conjunto ações emergentes, a partir dos estímulos sensoriais sobre a arquitetura.

Em resumo, apresentou-se uma arquitetura hierarquizada em três níveis para um AA do tipo móvel. Os níveis são: *reativo*, *instintivo* e *cognitivo*. Esses níveis possuem uma comunidade de agentes instintivos e nebulosos. Os agentes nebulosos estão implementados no nível reativo e interligam as percepções de proximidade com as ações motoras. Tanto os sensores para percepção como os motores da ação encontram-se distribuídos na arquitetura. A distribuição confere à arquitetura características da IAD (e outras áreas), como paralelismo, competição, cooperação, conflitos, etc. Os agentes do nível instintivo possuem esquemas internalizados sobre o nível reativo. Os esquemas ativados sobre a estrutura de agentes nebulosos são análogos aos esquemas biológicos [MAA95]. O enfoque da tese é muito próximo às arquiteturas comportamentais, onde as ações sucessivas de disparo visam um comportamento adequado ao ambiente. A seleção das ações ocorre sobre os agentes instintivos, a partir de dois sistemas classificadores² na arquitetura. O nível cognitivo habilita os agentes instintivos que podem estar ativos, objetivando estabelecer uma funcionalidade pertinente a um nicho. Uma ativação adequada nesse nível provoca uma *funcionalidade emergente* à arquitetura [Ste95b, Ste95a].

7.3 Resultados Alcançados

Esta tese apresenta uma proposta de arquitetura para agentes reativos do tipo móvel. A locomoção aqui alcançada é uma *funcionalidade emergente* de AA's [Ste94b], onde se presenciam um conglomerado de outros agentes internos mais simples, que de forma isolada conduzem a uma ação global. A *emergência* exibida não prescreve o rigor de: “*o inteiro é mais que a soma das partes*” [Baa93], mas apresenta o dinamismo e interações requeridas pelas partes [Yae93]. Os agentes internos apresentam conhecimentos locais e simples e, em momento algum, exibem predominância sobre um comportamento global. A arquitetura foi concebida a partir das partes menores (“*de-baixo-para-cima*” ou “*bottom-up*”). Um *comportamento emergente* (CE) é observado de uma articulação interna a uma ação externa [For90]. A partir da regularidade das ações por um período a “*longo prazo*”, faz com que um conjunto ações seja sintetizado em comportamentos regulares, emergentes em sua concepção. Os resultados obtidos nas simulações são seqüências de ações, que definem AA's dirigidos a nichos, com um CE em suas locomoções.

Como resultados decorrentes, citam-se algumas características adicionais à arquitetura apresentada:

1. A heterogeneidade de seus agentes internos. Alguns se utilizam de técnicas da lógica nebulosa e outros de regras de produção. Há uma integração de várias tecnologias de forma coerente e funcional (um dos objetivos da IAD);

²A existência dos seletores em projetos de AA's é fortemente justificada e reforçada pelo trabalho de P. Maes em [Mae91c].

2. Seu enquadramento dentro da visão de Solução Distribuída de Problemas da IAD, com característica de: raciocínio distribuído e funcionalidade global em seus níveis;
3. Sua estrutura hierárquica apresenta três níveis funcionais (reativo, instintivo e cognitivo), com as seguintes características: ausência de um modelo interno, flexibilidade, robustez, adaptação, dispensa operações em memórias e baixo custo computacional;
4. Seu nível reativo é composto por um conjunto de agentes nebulosos organizados, conforme uma topologia específica, que permite a implementação de conceitos como *competição*, *cooperação* e *conflitos* em arquiteturas distribuídas [Kle91];
5. Seu nível instintivo utiliza-se do conceito de *esquema motor* sobre um conjunto de agentes no nível reativo. Não é necessário um modelo interno para representar o ambiente. Há apenas os agentes instintivos que são ativados mediante certos contextos, utilizando-se para isso do conceito de *raciocínio baseado em contextos* [Tur93, Tur95];
6. Em um dos protótipos, seu nível cognitivo utiliza-se do conceito de “*autômato comportamental*” para caracterizar um comportamento de longo prazo ou recorrente. Os outros dois protótipos exibem autômatos comportamentais, mas implícitos em suas funcionalidades globais;
7. Há uma melhor compreensão e um avanço para o problema de AA’s, utilizando-se da forte inspiração biológica em seus níveis. Esse objetivo é continuamente enfatizado por R. Brooks [Bro91b] e pesquisadores da área.

A implementação dessa arquitetura permitiu a investigação de técnicas para resolução de conflitos em ambientes multiagentes, culminando na construção de um ambiente gráfico, destinado ao ensino dos cursos de IA, IAD e áreas próximas, como Vida Artificial, Agentes Autônomos, Computação Emergente [dO95], ..., etc.

7.4 Comparativo com Outros Trabalhos

Esta seção apresenta um paralelo entre esta tese e alguns trabalhos que se relacionam a AA’s. Esses trabalhos influenciaram e apontaram os limites inerentes a esta tese. Os enfoques variam das teorias de representação do conhecimento às redes neuronais, passando pela IAD e robótica. A lista encontra-se distante de ser completa, pois a área vem há tempo apresentando muitos enfoques dentro da história da IA. Os clássicos de R. Brooks [Bro91c, Bro91b, Bro91d] são sugestões de leitura na área.

A área de AA’s é debutante. Alguns pontos ainda estão por serem atingidos [Mae91a, Mae91b]. Primeiramente, compreender mecanismos neurofisiológicos de animais mais triviais, como insetos, moluscos, artrópodes, etc. O entendimento desses animais ainda é limitado, os enfoques são variados e a simulação matemática ainda

deixa a desejar. Portanto, há um caminho paralelo a ser percorrido por biólogos e roboticistas. Uma das boas tentativas nessa direção encontra-se na referência [RM93].

Na seção seguinte, há uma tentativa de “*varredura*” de trabalhos e paradigmas que de algum modo se relacionam com arquiteturas de AA’s. Os dois últimos tópicos dessa sequência não foram abordados nesta tese, mas possuem uma forte interseção na área de AA’s. Esses itens referem-se às questões da aprendizagem, um assunto que justificaria uma nova tese.

7.4.1 Lógica Nebulosa:

O nível reativo da arquitetura é uma rede de agentes nebulosos. Desse modo, trabalhos com estratégias e técnicas nebulosas são correlatos. Em face à rápida absorção dessa tecnologia pela indústria, possivelmente detenha o maior número de experimentos práticos na área de robôs móveis e controladores inteligentes.

O artigo de S.G. Kong e B. Kosko [Kos92] tornou-se um clássico dentro da área de sistemas nebulosos pela comprovação da robustez dessa tecnologia em relação a redes neuronais. O enfoque de estudo foi sob controle de sistemas adaptativos³. O problema abordado é o estacionamento de um caminhão em marcha-ré, ora com reboque, ora sem reboque. A robustez foi investigada sob critérios de sabotagem sobre as regras nebulosas, feita com a retirada de algumas regras de controle. Os sistemas nebulosos foram robustos até a remoção de 50% das regras de controle. Do conjunto inicial de 735 regras, reduziu-se a 435 regras. Mesmo com a eliminação, o comportamento sistêmico foi considerado aceitável. Esse exemplo dá uma noção de que o conhecimento de um sistema pode ser inicialmente modelado de forma aproximada. Adicionalmente, a prototipagem nebulosa é desenvolvida de maneira muito rápida e consistente.

Inicialmente, o problema é resolvido por técnicas nebulosas e neuronais, seguindo-se de uma abordagem comparativa, quantitativa e qualitativa de suas curvas de variáveis numa trajetória de estacionamento. Apesar da suavidade resultante das curvas do sistema neuronal, em longas trajetórias esse fator não é determinante. De modo resultante, foi constatado que o custo computacional é muito elevado em relação à abordagem nebulosa.

Na inexistência de uma função matemática que defina o par de dados entre entradas e saídas, *modelos livres de estimação*, as abordagens nebulosas e neuronais são interessantes em relação às técnicas tradicionais de controle. Nesse caso, os dados são obtidos a partir das associações do senso comum de um especialista ou, ainda, de representações estatísticas de um conjunto de amostras numéricas. Vale ressaltar que as abordagens tradicionais de controle apresentam problemas, como ruído, não-linearidade e dinâmica [FK94].

Para S.G. Kong e B. Kosko [Kos92], a disponibilidade dos dados de controle

³É considerado um controle adaptativo à mudança de um ponto da saída segundo uma superfície espacial onde se define um comportamento sistêmico. Exemplificando, no caso de duas entradas e uma saída, gera-se um mapeamento tridimensional.

do sistema real determina a abordagem do sistema controlador, seja ele nebuloso, neuronal, matemático, híbrido, estatístico⁴, etc. Na indisponibilidade de dados sobre o sistema, uma estimativa inicial de bom senso é aceita. Nessas estimativas iniciais de controle, um esquema de regras nebulosas é factível como uma primeira abordagem.

No trabalho de S.G. Kong e B. Kosko, a definição dessas regras foi feita pela aglomeração de pontos próximos sobre os conjuntos de entrada e saída. Cada variável possui um conjunto específico de pontos, especificando um *universo de discurso*. Uma aglomeração delimita regiões sobre os conjuntos de pontos, cuja idéia final é mapear todos os dados do sistema. Um termo nebuloso associa uma região em particular, sobreposta ou não a outras regiões. As regras por sua vez são construídas considerando as variáveis que controlam o sistema e seus respectivos conjuntos de termos. Sendo assim, um conjunto de associações é estabelecido entre os termos da(s) entrada(s) e saída(s). Esse esquema de mapeamento é denominado de regras *nebulosas de memória associativa* (**FAM**: “fuzzy associative memory”). O esquema FAM é largamente utilizado pelos trabalhos na área de controle nebuloso. O método de inferência utilizado sobre as regras FAM é o da *correlação mínima* (operador “*mín*”). O método empregado na desfuzzificação é do *ajuste pelo valor máximo*, comumente referenciado como centróide nebuloso. Detalhes sobre outros métodos de cálculo para regras nebulosas podem ser encontrados em [TS87, Lee90a, Lee90b, Kli88].

O controle nas abordagens neuronal e nebulosa tem seus resultados sumarizados em:

- Os sistemas neuronais são mais precisos, mas de alto custo computacional. Algumas vezes seus resultados podem ser comprometidos em face de um pequeno número de amostras, ou ainda, de uma heterogeneidade elevada, evitando assim o poder de síntese (generalização) que elas exibem;
- Os sistemas neuronais apresentam um processamento típico de “caixa-preta”, fator que pode dificultar detalhes de valores presumidos na etapa de teste;
- Os sistemas nebulosos permitem um refinamento e ajustes sobre as regras FAM, com facilidades de considerar as “regras de polegar” de um especialista. As inclusões ou exclusões das regras são realizadas com maior facilidade;
- Os sistemas nebulosos são compreensíveis quanto a sua estrutura gerada. Há uma rápida prototipação do sistema, que permite uma validação construtiva do conjunto estruturado de regras FAM.

Algumas considerações de [Kos92, FK94] nortearam a escolha da lógica nebulosa como ferramenta na rede de agentes do nível reativo. A diferença é que a maioria dos sistemas de controle nebuloso possuem mecanismos centralizados nas regras inferenciais. O problema não é decomposto, e as variáveis das entradas e saídas do sistema são associadas “*par a par*”. As regras são construídas segundo uma semântica de

⁴Com uma regressão linear, tem-se o histórico do processo.

alto nível, analogamente a formulação de um conhecimento por parte do especialista. As regras sofrem aglomerações, simplificações, restrições, etc, mas dispostas em uma base centralizada, o que leva a uma perda do valor semântico das regras. No intuito de preencher todos os espaços de estados comportamentais do sistema, detecta-se a ocorrência de regras redundantes ou replicações. Eventualmente, um problema inverso pode ocorrer, isto é, ausência de uma completude.

Esse prejuízo semântico foi evitado nesta tese pela decomposição da relação direta existente entre os conjuntos das entradas e saídas. A fragmentação, análoga à SDP (ver capítulo 3), deu origem a uma rede de agentes nebulosos. Essa cisão fez com que o número de variáveis a serem computadas aumentasse. Contudo, esse aumento é compensado pela distribuição das tarefas, que tem conseqüências diretas, como uma maior controlabilidade das regras, depuração e diagnósticos imediatos, menor custo de aprendizagem das regras, etc. Finalmente, o maior ganho é na compreensão semântica do sistema, uma vez que o conhecimento encontra-se “*diluído*” sob forma de uma rede de agentes. Tem-se como resultante uma semântica embutida, que segue desde o nível de agente isolado até a sua rede cooperativa.

Quanto à questão topológica da rede, ela seguiu as idéias de circuitos neuronais biológicos com mensagens inibitórias e excitatórias, com seus possíveis esquemas funcionais [RDBS91]. A principal diferença da arquitetura desta tese com os trabalhos na área de agentes nebulosos [Sug85, Iwa93, SV93, Tsc94, SW95] é a decomposição que ocorreu no mapeamento sensorial-motor. O resultado é uma rede de agentes nebulosos (análogo às “*microteorias*”, segundo B. Huberman [Wel91]) de topologia fixa.

Na modelagem da base de conhecimento dos agentes nebulosos, utilizou-se de uma idéia contrária a de Kong e Kosko. Nesta tese, as regras foram incorporadas gradativamente, seguindo apenas um bom senso. No momento em que o AA não apresentasse uma inferência nula, o procedimento de inclusão de novas regras era interrompido. Após a ocorrência de uma inferência nula, localizava-se o agente que provocou tal inconsistência, para então acrescentar uma nova regra nebulosa à base já existente. Por meio desse procedimento, cada agente deve apresentar um número reduzido de regras, do número máximo permitido (produto cartesiano entre todos os termos nebulosos). Contudo, esse número reduzido não é um conjunto mínimo de regras, o que seria muito complexo dentro da lógica nebulosa. Algumas referências adicionais sobre o paradigma da lógica nebulosa são encontrados em [Mam85, TS87, KH87, Kli88, Lee90a, Lee90b, Zad94].

7.4.2 Visão Conexionista:

Um projeto representativo na área de móveis autônomos (MA's) é desenvolvido por U.R. Zimmer e outros [vP94c, Zim94, vP94a, vP94b]. Esses trabalhos são centrados sobre dois projetos físicos de rôbôs: o MOBOT e o ALICE. As linhas de pesquisas desses estudos seguem da captação sensorial, modelagem do ambiente, aprendizagem em tempo real, seleção de caminhos e controle sobre os motores atuadores. Devido

ao número de detalhes relacionados aos MA's, uma descrição é aqui omitida.

Sucintamente, os MA's possuem sensores de alta resolução, tanto na detecção de objetos, como na captação de sinais emitidos pelo ambiente. Por exemplo, a detecção de um gradiente de luz, fornecendo a distância entre um ponto e o móvel. Há um esquema de localização global, com um mapa interno, em que o MA procura reconhecer locais não visitados. No encaminhamento sobre espaços vazios é construído um mapa reticulado sobre essas áreas. O reconhecimento dos objetos é feito com redes neuronais auto-organizáveis de Kohonen [Koh84]. Essas redes dinâmicas sofrem um processo “crescimento” e “emagrecimento” em seus estados, considerando o mapa corrente e os sinais dos sensores. O mapa é construído com base em duas técnicas: a *geométrica* (ou exata) e a *topológica qualitativa*. Uma vez construído o mapa, o módulo de pilotagem utiliza-se do algoritmo A* para realizar uma navegação entre dois pontos.

Com relação às diferenças e às igualdades dos trabalhos de U.R. Zimmer e outros [vP94c, Zim94, vP94a, vP94b] e esta tese tem-se:

- O projeto MOBOT e o ALICE são de alto custo computacional (memória, disco, processador, etc). Contudo, realizam façanhas como a aprendizagem de ambientes estáticos em tempo real;
- O uso de sensores precisos que estimam posições globais;
- As saídas são fornecidas em termos de um ângulo de desvio (uma direção) e um deslocamento;
- O projeto ALICE gera mapas confiáveis, permitindo uma navegação segura com conhecimentos acumulativos do ambiente. Nesta tese, a navegação é baseada apenas nas entradas sensoriais, não tendo nenhum modelo do ambiente disponível.

Em resumo, observa-se que um eventual comparativo é relativamente desigual, considerando recursos e objetivos.

7.4.3 Visão da IA Simbólica:

Um sistema simbólico é definido como um conjunto de entidades físicas (símbolos) que podem ser combinados para formarem estruturas, são habilitadas a executarem processos sobre esses símbolos, de acordo com um conjunto de instruções simbolicamente codificadas. Essa definição é conhecida como “*hipótese dos sistemas simbólicos e físicos*” de A. Newell e H.A. Simon [WJ94], sendo uma condição necessária para um sistema executar uma ação inteligente.

A partir dessa definição, J. McCarthy introduz o conceito de *agente deliberativo* ou *automação de processamentos sentenciais*. Um agente deliberado possui um modelo simbólico do mundo explicitamente representado. Suas decisões (por exemplo, as ações a serem executadas) são feitas por um raciocínio lógico.

M.J. Wooldridge e outros [WJ94] apontam dois pontos a serem resolvidos segundo essa visão de agente simbólico. Sobre esses pontos, um certo encaminhamento foi investigado nesta tese:

O problema da transdução: reflete a tradução do mundo real de modo preciso e adequado a uma descrição simbólica num tempo factível;

O problema do raciocínio na representação: reflete a maneira como inferir novos dados ou interpretá-los corretamente, a partir de entidades complexas (simbólicas), que representam o mundo real e seus processos, ou seja, como o agente realiza um raciocínio sobre uma representação em um tempo hábil.

A maioria dos pesquisadores aceita que os problemas acima estão longe de uma solução, igualmente, dentro de áreas como a visão, o entendimento de linguagem natural, o reconhecimento de imagem, a aprendizagem, etc. Mesmo para problemas aparentemente triviais, como o raciocínio de senso comum, têm se tornado extremamente complicados [WJ94]. Contudo, a comunidade da IA simbólica exerceu alguns esforços em direção a projetos de agentes, como os *agentes planejadores*.

O planejamento é essencialmente uma programação automatizada, um projeto detalhado de um conjunto de ações, que resulta em um objetivo desejado. Um dos planejadores mais conhecido é o STRIPS [BF81, Lau90, RK94]. Seu algoritmo era muito simples e ao mesmo tempo mostrou-se ineficiente para problemas de complexidade moderada. Um dos principais resultados na área de planejamento deve-se a D. Chapman. Ele estabeleceu importantes resultados teóricos, que indicam que mesmo em técnicas refinadas o planejamento está sujeito a ser descartado em sistemas com restrições de tempo [WJ94]. Outro resultado foi a constatação que a maioria das ações desenvolvidas ao longo de um processo não passam de repetições ou rotinas com ligeiras variações em torno de uma linha básica [AC91]. Esses resultados influenciaram diretamente os pesquisadores da IA clássica, que buscaram alternativas em outras abordagens, deixando o planejamento em segundo plano.

Considerando as questões levantadas acima sobre a IA simbólica, a arquitetura proposta nesta tese apresentou a seguinte relação:

- Não há uma captura exata do ambiente em que o AA se encontra. A percepção sensorial é aproximada e mínima, isto é, há poucos elementos;
 - Não há uma construção perfeita de um modelo interno ou representação sobre o ambiente a partir dos dados perceptuais. A exemplo da formiga caminhando na praia, há sim uma estrutura interna de reflexos aproximados, em relação ao ambiente. O modelo de raciocínio é aproximado e se utiliza de uma representação típica das regras de produção no nível instintivo;
 - Não há um planejador. O enfoque é dirigido a nicho, a exemplo do repertório limitado exibido de alguns animais. Na arquitetura apresentada, há uma concepção em nível cognitivo de uma meta a atingir [HR95].
-

7.4.4 Visão de Esquemas:

“Esquemas são pacotes de ações que deveriam ser executadas conjuntamente, em algumas situações para atingir um objetivo do agente, ou realizar alguma ação. Na psicologia e biologia, esquemas são idealizados (concebidos) para serem aprendidos pela experiência repetitiva. Esquemas, então, permitem ao agente fazer previsões, baseadas em experiências passadas sobre o que deveria ser feito em seguida. Uma vez aprendidos, os esquemas podem ser aplicados a novas situações, considerando-os como um programa que pode ser executado sem ter que repensar o seu comportamento” [Tur95].

De acordo com essa citação introdutória, os esquemas são adquiridos por sucessivas repetições de um evento, que uma vez aprendidos tornam-se automáticos em algum nível de consciência. Basicamente, tais recorrências entre estados ocorrem em um *nível motor*. Após sucessivas ocorrências ou a detecção de uma relevância, um esquema é elaborado em um *nível cognitivo*.

Há vários tipos de esquemas. Os mais conhecidos na neurociência, na psicologia e robótica, são: *perceptual*, *motor* e *“script”*. Resumidamente, o esquema perceptual incorpora as montagens das “rotinas” de reconhecimento de padrões e as entrega como informação perceptual base, necessária ao esquema motor, que ocorre pela ativação funcional sob uma estrutura física. No caso do agente, é a resposta para um estímulo do ambiente [MAA95, Ark93]. O esquema *“script”* representa a dinâmica de um cenário com um conjunto de ações estereotipadas pelo agente, que normalmente se encontram “montadas” em um nível cognitivo. Sendo assim, um esquema do tipo *“script”* é composto por um conjunto de ações a serem executadas. Por exemplo, em um *“script”* de guiar um carro, há uma ação de dar a partida no carro, outra para engrenar, etc.

Os trabalhos de R.M. Turner [Tur95, Tur93] utilizam-se do raciocínio baseado em esquemas para controlar a navegação de um Veículo Autônomo Subaquático (VAS). O protótipo construído opera em ambientes aquáticos, que apresentam características como: complexidade, imprevisibilidade e hostilidade. Desse modo, para que o VAS apresente um comportamento inteligente, deve ser *sensível a todos os contextos*. O contexto⁵ modula o comportamento do VAS, que é afetado nas ações correntes, contudo, considera fatos como objetivo da missão e limitações inerentes.

Sob esse cenário, foi elaborada uma biblioteca de quadros (*“frames”*), que descreve todos os possíveis contextos. Assim sendo, uma representação de conhecimento típica ao paradigma de quadros foi usada para descrever os *esquemas de contextos*. Nos quadros são considerados vários parâmetros, como velocidade, posição, detalhes da missão para o esquema em questão, etc. Os *esquemas de contextos* são esquemas procedimentais de ações a serem executadas, análogas a planos e regras.

A arquitetura de R.M. Turner possui um gerenciador de contexto, que recupera na memória o melhor esquema para a situação corrente. Deve haver uma casamento

⁵O ambiente, outros VAS na área, bases submarinas, piloto, etc.

entre um esquema contextual e o estado corrente. Uma vez selecionado o esquema, ele desencadeia um conjunto de ações que vigoram enquanto o contexto for compatível.

Os trabalhos de R.M. Turner apresentam uma forte interseção com essa tese nos seguintes pontos:

- O ambiente é desconhecido e potencialmente infinito quanto aos valores nas entradas;
- O VAS apresenta o conceito de missão a cumprir. A missão é um fator ponderável dentro da biblioteca de esquema. Os agentes apresentados nesse trabalho possuem nichos comportamentais bastante restritos;
- O VAS possui um selecionador de contexto. R.M. Turner pretendia usar um seletor nebuloso, que não foi implementado, na escolha dos contextos. Essa técnica foi adotada nesta tese com sucesso;
- O VAS utiliza-se do conceito de esquemas do tipo “script”. Por outro lado, nesta tese enfatizou-se o uso de esquemas motores;
- O VAS apresenta uma estreita relação com o ambiente, bem como a arquitetura proposta nesta tese. Apesar de um “script” poder ser de longa duração, o monitoramento é constante e paralelo a todas as entradas.

As representações dos contextos com o uso de “frames” simbolizam situações generalizadas do tipo problema-solução. As características são filtradas de vários ou muitos casos semelhantes no passado. Sob essa ótica, o raciocínio baseado em contextos, usado por R.M. Turner, é visto como uma forma generalizada do raciocínio, baseado em casos. Esse detalhe não se aplica a esta tese.

Outro trabalho significativo nesta linha tem sido feito por R.C. Arkin [Ark91]. Refere-se à navegação de robôs físicos, baseada em esquemas. Em [Ark93], um esquema motor é gerado para comportamentos primários, que refletem reações do tipo estímulo-resposta. Muitos comportamentos são necessários à realização de uma tarefa. Essa abordagem é contrária à adotada nesta tese, em que existe uma arquitetura fixa, e os esquemas são desenvolvidos sob esta estrutura. Em [Ark93], não há tal estrutura física. Os esquemas motores são elaborados, mas formados a partir de comportamentos requeridos ou desejados. Alguns esquemas gerados são do tipo: mover-à-frente, evitar-obstáculo-estático, estacionar, seguir-o-líder, seguir-para-um-objetivo, etc. A representação construída desses esquemas é o *gradiente potencial*, uma idéia fundamentada nas *representações analógicas* de L. Steels [Ste91].

Finalmente, algumas conclusões feitas por R.C. Arkin [Ark93] motivam uma reflexão sobre o uso de esquemas, pois:

- Possibilitam que processamentos perceptuais sejam mais eficientes, uma vez que há um direcionamento para se reconhecer certos padrões, em vez de buscar objetivos sobre um modelo de mundo;
-

- Permitem o reuso de software;
- Suportam a computação paralela;
- Permitem um desenvolvimento incremental, uma vez que são altamente modulares;
- Permitem gerenciar a incerteza da percepção com as novas amostras freqüentes do ambiente;
- Permitem respostas em tempo real, uma vez que a computação motora é econômica.

Há um encorajamento final, que mesmo não dispondo da descrição de um sistema animal, tais abstrações esquemáticas devem ser propostas na construção de sistemas inteligentes. As abstrações de esquemas desempenham um papel fundamental em uma primeira abordagem, e refinamentos sobre esses modelos são feitos de acordo com as experimentações.

7.4.5 Visões Alternativas à IA:

Englobamento (“Subsumption”): esse tipo de arquitetura encontra-se inserida na categoria das reativas, cuja visão é a mais antagônica em relação à IA tradicional. O englobamento influenciou a maioria dos projetos de agentes reativos (ver capítulo 3). Contudo, um agente reativo não necessariamente precisa ser construído sob o paradigma do englobamento, mas, tem predominado na maioria das implementações físicas de AA’s [dS96, Bro86, Bro91c, Bro91d]. Suas relações com esta tese são as seguintes:

- Os sinais da percepção se conectam de modo independente aos módulos comportamentais. Nesta tese, os comportamentos são definidos por esquemas a serem ativados sob uma estrutura motora;
- Os módulos estão dispostos em camadas, mas não há uma hierarquia definida quanto a seus fluxos de controle (sinais *supressores* e *inibidores*) na ativação de um comportamento em particular. A visão do projetista busca especificar uma propriedade emergente do agente. Esse procedimento é análogo ao nível cognitivo desta tese, quando se seleciona um subconjunto de agentes no nível instintivo;
- Não há um número limitado de camadas comportamentais. Apesar de simulações chegarem a sete camadas, nas experimentações físicas esse número ficou em três [Bro91c]. Sem relação alguma, coincidentemente, esse é o número de níveis da arquitetura proposta;
- Não há um classificador de módulos para atuarem segundo um contexto. Todos os módulos visam as suas próprias ações, suprimindo e/ou inibindo⁶

⁶Entre os módulos comportamentais há dois tipos de sinais: os *supressores* e os *inibidores*. Desses sinais, deu-se a origem do nome desta arquitetura: “*subsumption*”.

as ações dos módulos das camadas inferiores. Esse aspecto provoca um paralelismo muito interessante à arquitetura, detalhe também observado no nível reativo da arquitetura proposta;

- Os módulos comportamentais possuem ligações fixas entre si, lembrando um autômato finito estendido (AFE). O nível cognitivo do protótipo Maja pode ser visto como um AFE;

O englobamento tende a explorar as propriedades físicas do agente e do ambiente. A intensa interação entre agente e ambiente é um requisito fundamental à observação de um comportamento inteligente. Esse acoplamento e interação agente-ambiente não puderam ser explorados nesta tese, em face de seu desenvolvimento ter sido em ambiente simulado.

“TouringMachines”: em sua tese de doutorado, I.A. Ferguson [Fer92] desenvolveu uma arquitetura híbrida de um agente simulado. A arquitetura possui três camadas de controle, embutidas em uma *estrutura de controle*, que pondera a comunicação entre as camadas com os subsistemas de percepção e ação interfaceados com o ambiente. Cada camada é independente das demais e executa processos concorrentes. A primeira camada é a *reativa* e destina-se a responder rápidos eventos restritos no tempo, evitando, sempre que possível, um acesso às demais camadas. Sua implementação foi feita com regras do tipo situação-ação, isto é, um estilo próximo ao englobamento de R. Brooks. A camada de *planejamento* constrói planos e seleciona ações para serem executadas, considerando o objetivo do agente. Os planos são parcialmente pré-elaborados, mas há um mapa topológico do ambiente previamente conhecido. A última camada é a de *modelagem*, que contém representações simbólicas do estado cognitivo. Trata de questões, como conflitos no alcançar objetivos, inferências inesperadas, mediar conflitos entre as duas outras camadas, etc. A *estrutura de controle* da arquitetura foi implementada com *regras de controle* do tipo “*se-então*”. A “TouringMachines” com relação a esta tese apresenta as seguintes características:

- Possui um esquema de controle centralizado para as três camadas;
- A existência de um mapa do mundo para construção de planos;
- Possui uma camada reativa modelada num estilo próximo ao englobamento;
- A camada de modelagem é próxima ao nível cognitivo do protótipo Maja;

O número de níveis da arquitetura “TouringMachines” e esta tese são iguais, contudo, a funcionalidade dos mesmos exibe algumas diferenças. O nível cognitivo desta tese e a camada de planejamento da “TouringMachines” definem as metas que o agente deve cumprir. Essa é a maior proximidade funcional.

7.4.6 Visão quanto à Aprendizagem:

A aprendizagem é um dos temas mais importantes e enfocados dentro da IA. Em uma das definições de inteligência o interessante é que o programa aprenda novos fatos e que isso conduza à mudança comportamental. Há várias teorias sobre o tema. A mais conhecida é a *aprendizagem indutiva* [RK94]. A indução procura definir um padrão a partir de exemplos que lhe são apresentados. Esse tratamento simbólico objetiva uma classificação de exemplos, visando distribuí-los em classes características, obtidas por analogias, diferenças ou até mesmo estatisticamente. Após essa classificação, o novo conhecimento é incorporado ao programa.

Uma outra classe de metodologias pertinentes à aprendizagem é a *aprendizagem por recompensa* (“*reinforcement learning* - *RL*”). Há muitas variações em torno do RL. Um resumo sobre é encontrado em [Mat95, Mar96]. Contudo, o modelo de RL que tem apresentado boas perspectivas a robótica usa o algoritmo “*Q-Learning*” [MT92, Thr94, Pet95, Mer96, PM96]. No RL, o agente se adapta com base na realimentação externa que recebe do ambiente, de acordo com a interpretação feita a escalares de recompensas positivas ou negativas. Ao longo do tempo, o objetivo do agente é maximizar as recompensas e minimizar as punições. O agente ao realizar uma ação tem como retorno do ambiente uma nova situação e o valor da recompensa. A maneira como se realiza um mapeamento dos estados é chamado de *estratégia*, isto é, uma associação entre a situação e a ação. O sinal de recompensa é produzido como consequência direta ou indireta do uso dessa associação. Em suma, o RL busca de algum modo uma convergência em torno da repetição de um número de eventos⁷, de modo que o ambiente seja restrito em algum sentido. Por exemplo, um ambiente com um mapeamento gradeado permite identificar um número limitado de instâncias que cada célula em torno do agente possa assumir: ocupada ou vazia. A principal desvantagem do método é o número de experimentos necessários. O número é muito elevado em relação a que um animal real necessita para caracterizar um novo comportamento [Bro92]. Um outro problema é a segmentação necessária em pequenas partes de um comportamento maior. O agente submetido ao processo de aprendizagem considera um seqüenciamento de estímulo-ação, sem a caracterização de comportamentos mais específicos. Essa cisão em comportamentos menores reflete o detalhe que permite o agente ser submetido a qualquer tipo de ambiente. Finalmente, em [Ste94b] Luc Steels aponta algumas dificuldades adicionais ao RL quando submetido às experimentações físicas:

- Determinação do sinal de recompensa: é irreal pensar que o agente incorpore com fidelidade o novo sinal de recompensa após uma ação ou uma série de ações;
- Determinação de uma associação viável: o RL assume uma pesquisa de tentativa e erro para encontrar uma boa associação. A associação pode ser de difícil descoberta, a menos que o agente já esteja próximo a um comportamento desejado;

⁷Apesar de suportar um número infinito de eventos.

- Determinação da atribuição de crédito: geralmente, o agente possui vários sensores e pode estar realizando várias ações simultaneamente. A questão é como atribuir uma relevância a cada par de uma interpretação sensorial a uma ação.

Por outro lado, os animais exibem uma inata habilidade de aprendizagem restrita a classes particulares de comportamento a partir de uma estrutura interna já formada [Bro92]. Apesar dos animais apresentarem um número limitado de comportamentos, a aprendizagem ocorre numa extraordinária velocidade de convergência, sob estruturas físicas que estão além de programas numéricos [Mat95]. O RL trabalha robustamente sobre problemas simples, mas quando há uma variedade de dimensões (domínios complexos) a serem contabilizadas, o método ainda não se encontra completamente amadurecido [Ste94b]. Enfim, a aprendizagem de novos comportamentos em experimentações físicas de AA's tem se mostrado um assunto de grandes desafios e dificuldades [MB90, Bro92].

7.4.7 Visão Evolucionista:

Uma outra abordagem de aprendizagem tem se definido junto à Vida Artificial (VA) [Lan92, Pra94], mais especificamente utilizando-se dos Algoritmos Genéticos (AG's) [Col95, HP95, SS96]. Os AG's encontram-se sob o tema “Evolução” na VA. Há outros paradigmas evolucionistas, apresentados em [Lan92, Bit96].

Resumidamente, as técnicas dos AG's se baseiam na *Teoria da Seleção Natural*, proposta por Charles Darwin. Em AG's, um genótipo é representado por uma cadeia de caracteres ou de bits, codificando ou representando uma solução potencial a um problema. Por exemplo, uma cadeia de caracteres pode codificar a matriz de pesos de uma rede neuronal ou, ainda, uma tabela de transição de uma máquina de estados finitos [Lan92]. Esses caracteres são “convertidos” (traduzidos, instanciados) para fenótipos via um interpretador específico do problema. As trocas internas da sequência de caracteres dos genótipos são realizadas segundo os operadores típicos da genética. Os operadores genéticos são: a *reprodução*, a *mutação*, o *cruzamento*, a *inversão* e a *duplicação*. As descrições desses operadores podem ser encontradas em [Lan92]. A cada novo genótipo há um fenótipo correspondente. O interpretador funciona como um selecionador que visa encontrar a descendência de uma geração mais apropriada à solução do problema. No caso da rede neuronal, o novo conjunto de pesos deve ser instanciado à rede, que ser testada com os valores das entradas e saídas. O fenótipo é a instanciação física dos pesos na rede. O processo é iterativo entre um algoritmo, que realiza “trocas” (rearranjos) e uma instanciação física do problema. Dois pontos importantes são derivados desse processo: a existência de um *algoritmo genético* básico e uma *função de avaliação*.

O *algoritmo genético* preocupa-se basicamente com a criação ou reprodução das populações de genótipos. A descrição dos passos desse algoritmo pode ser encontrada em [MF93, Bit96]. Quanto à *função de avaliação*, ela deve ser capaz de avaliar a aptidão de cada fenótipo. Essa função não precisa deter o conhecimento sobre como encontrar uma solução para o problema, mas deve atribuir uma nota ao desempenho

de cada fenótipo da população gerada [Bit96]. O valor é representativo de quanto esse fenótipo corresponde ou se ajusta a uma solução do problema. Desse fato, a função de avaliação também é conhecida por “*fitness function*” (função de ajuste ou de concordância) de um domínio do problema.

O algoritmo genético é de fácil implementação. Porém, o mesmo não acontece com a função de ajuste. Retornando-se a um AA em um ambiente com várias variáveis, num processamento concorrente e paralelo, a questão reside em como definir essa função de ajuste. Existe uma função ajuste invariável no tempo?

Antes que técnicas evolutivas como AG’s sejam aplicadas a experimentações físicas de AA’s, algumas considerações são feitas a priori:

- Há uma grande diferença entre a teoria dessas técnicas e um AA real. A interação dinâmica com o ambiente não é contabilizada em simulações como AG’s;
- A estrutura da pesquisa de espaço de possíveis programas ou comportamentos é muito dependente da representação utilizada pelos AG’s. Em [Bro92] é recapitulado um estudo de um agente simulado⁸. Ele usa um AG diretamente sob uma estrutura tipo Lisp, em vez da representação tradicional de uma cadeia de caracteres ou bits.

Em face às considerações acima, há uma tendência em desenvolver os AG’s em representações celulares de espaço, buscando de imediato uma eficiência e um baixo custo computacional. Em experimentações físicas de AA’s, as representações celulares apresentam três problemas. Primeiramente, não existe a noção de incerteza presente no mundo real. Em segundo, há uma tendência de rejeitar as informações perfeitas retornadas do sensor, pois confunde-se a visão real do mundo com a do agente. Em terceiro, a dinâmica do agente simulado tende a ser mais frágil do que no mundo real, onde existe o ruído e o processo estocástico, que suavizam os acontecimentos. Enfim, há uma “distância a ser quebrada” pelo AG de um mundo simulado à experimentação física. Em resumo tem-se: “... *enquanto o mundo simulado de algum modo é mais simples que o mundo real, eles são paradoxalmente algumas vezes mais difíceis para se trabalhar*” [Bro92].

Entre as conclusões apresentadas por R.A. Brooks sob a visão evolucionista com AG’s e a robótica, talvez a mais importante é a certificação promissora das técnicas de programação genética (PG). A PG envolve áreas como: AG’s, Computação Evolutiva, ... [Bit96]. Os AG’s são destinados a construir programas baseados em comportamentos, que podem ser executados em robôs reais. A idéia foi originalmente lançada por Christopher G. Langton, em 1991, e ratificada por R.A. Brooks em [Bro92].

Finalmente, retomando e respondendo a questão da *função de ajuste*, as seguintes direções são apontadas:

⁸A arquitetura usada nestas simulações resultou em várias simplificações. O pionismo desses trabalhos deve-se a John R. Koza [Koz90, Koz91].

- O genótipo não deve ser elementar, isto é, a semente deve ir além de uma sequência de caracteres. A questão é encaminhada para o *Paradigma da Programação Genética*, onde os genótipos são expressões de uma linguagem de programação;
- O uso de técnicas para compressão dos genótipos. Os genótipos acima descritos geram estruturas em árvores, que podem se ramificar e tornar os cálculos razoavelmente complexos;
- O uso da função de ajuste pode ser progressiva a partes do agente. Por exemplo, o AG considerando apenas o conjunto frontal de sensores;
- O fatoramento de comportamentos complexos com o objetivo de acelerar a aprendizagem e diminuir as buscas de estados. Essa técnica foi utilizado em [Col95], onde várias configurações de *sistemas classificadores*⁹ foram incorporados à arquitetura do agente, objetivando um melhoramento no desempenho do AG.

Em resumo, a combinação com outras técnicas, que visem acelerar o processamento do AG, torna-se um dos principais caminhos a ser seguido pela Computação Evolutiva. Uma sinopse das várias aplicações dos AG's e sua sua interseção com a VA são encontrados em [MF93].

7.5 Trabalhos Futuros

1. Implementar fisicamente a arquitetura proposta ou parte desta. Alguns conceitos de controle inteligente de processos são observados no nível reativo. Por exemplo, o fato desse nível ser inerentemente adaptativo;
2. Alterar o comportamento de “vaguear” (passeio aleatório) no nível cognitivo do protótipo Maja. Esse estado ocorre cada vez que o agente se encontra em um beco sem saída e tem a necessidade de abandonar temporariamente o seu objetivo. A proposta é melhorar substancialmente a maneira do agente realizar o passeio aleatório. A idéia é prover o agente de alguma heurística quando se encontrar no estado de “vaguear”. Essa heurística é descrita por um outro autômato comportamental, apresentado na figura 7.1.

O cerne da heurística acima é que o agente caminhe em círculos cada vez maiores. Ora o agente caminha junto à parede esquerda por um período $\approx t_1$ e em seguida muda para o lado direito por um período $\approx t_1$. O detalhe é que com o passar do tempo, o agente vai se afastando cada vez mais da sua região de origem, mas de maneira ordenada. O tempo t_2 é gasto para o agente cruzar

⁹Ver [Zan90, Far90].

a sala ou o corredor. Eventualmente, não precisa ser contabilizado o tempo no estado de vaguear. Desse modo tem-se:

$$t_{vaguear} \cong 2.t_1$$

Em resumo, o agente acompanha as paredes pelo seu lado esquerdo (direito) por um período de tempo $\approx t_1$, para em seguida atravessar o ambiente e acompanhar as paredes pelo seu lado direito (esquerdo). Essa heurística não foi implementada. A dificuldade consiste em adaptar e reusar parte do código do agente Brooks, habilitado à esquerda e à direita, para os respectivos estados da figura 7.1;

3. Investigar a aprendizagem sob a arquitetura descrita. O tema aprendizagem evolucionária encontra-se sob investigações preliminares na área de AA's. Um interessante trabalho é alterar parcialmente a arquitetura proposta para incorporar uma habilidade de aprendizagem;
4. Avaliar, simplificar e implementar o modelo genérico do agente cognitivo proposto por G. Bittencourt [Bit97], visando um agente reativo dirigido a nichos. Em [Bit97], G. Bittencourt propõe uma generalização teórica que incorpora a arquitetura desenvolvida nesta tese, objetivando um modelo para um agente do tipo cognitivo. O modelo desse agente está baseado na hipótese de três atividades cognitivas: *auto-organização*, *natureza evolucionária* e *dependência histórica*. A arquitetura possui os três níveis: *reativo*, *instintivo* e *cognitivo*. O nível reativo é composto por uma rede neuronal no seu campo perceptivo. No campo motor há um sistema nebuloso adaptativo. Entre esses dois campos,

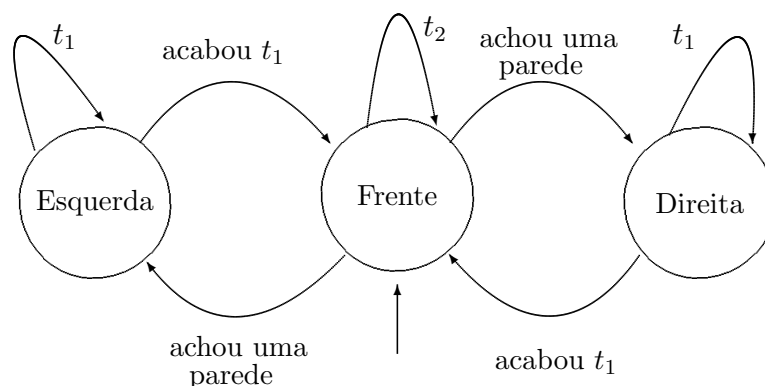


Figura 7.1: Comportamento Itinerante com Orientação

há um sistema evolucionário, como um AG. Ele gera seus genótipos (sequência cromossômica), visando a alteração dos pesos da rede neuronal, bem como a mudança nas funções de pertinência do sistema nebuloso. Essa é uma primeira solução do item anterior. Novamente, depara-se com a definição da *função de ajuste*. No nível instintivo há um módulo que realiza uma aprendizagem indutiva a partir dos melhores genótipos, objetivando a geração de modelos ao nível superior. O nível cognitivo realiza um tratamento simbólico a partir desses modelos, gerando representações factuais por meio de uma *lógica de transformação dual* [Bit97]. Os fatos retornam ao nível instintivo, passando por um codificador genético, que converte os fenótipos em genótipos de novas populações mais ajustadas. Os novos genótipos visam uma definição e adaptação da *função de ajuste*. O ponto central é a aprendizagem sobre a função de ajuste, que exhibe conceitos de “emoções” e “sentimentos”, a partir dos níveis reativo e cognitivo, respectivamente. Observa-se que a abordagem evolucionária apresenta-se como forte candidata a resolver questões de aprendizagem, bem como propõe resolver a questão da *função de ajuste* (um dos focos centrais da proposta), de modo evolutivo e adaptativo. Enfim, essa proposta é muito promissora, vislumbrando uma solução para o ponto levantado no item anterior;

5. Avaliar o reuso conceitual dos AA's, mais especificamente a partir da arquitetura proposta, nos Agentes de Software (AS's) [Mae95]. Os AS's são programas computacionais que capturam as necessidades e interesses do usuário (com interfaces sofisticadas, como voz e imagem) com objetivo de encaminhar um processo autônomo a essa tarefa. Nesse momento, P. Maes [Mae95] discute a fusão entre a IA e Teoria da Evolução, promovendo uma evolução artificial de softwares, num limite de uma “*auto-programação*” para atingir um determinado objetivo. Uma vez mais, esses agentes devem apresentar características, como robustez, adaptação, aprendizagem da experiência e respostas a situações imprevisíveis com um novo repertório de diferentes métodos. Exemplos e perspectivas da área de AS's encontram-se em nosso cotidiano de tarefas com um computador. Por exemplo, um agente que acompanhe as mensagens de chegada num correio eletrônico e as classifique em ordem de importância desejada pelo usuário. Um outro exemplo é o agente que realiza uma pesquisa em vários nós de uma rede de computadores. De modo análogo, essa foi uma das funcionalidades exibidas pelo protótipo Steels, que percorria uma lista de nós num ambiente desconhecido, cuja função de ajuste era a minimização angular entre o agente e o nó atrator. De modo oportunístico, P. Maes [Mae95] vislumbra um casamento entre “Turing” e “Darwin”, no sentido que esses softwares evoluam num processo espaço temporal baseando-se no interesse do usuário. Nesse sentido, o modelo de agente cognitivo proposto por G. Bittencourt [Bit97] preenche tais requisitos. Outro importante trabalho deve-se a O. Etzioni [Etz93], que propõe o uso dos “*softbots*” (“*software robots*”) sob domínios específicos, como um sistema operacional ou um banco de dados. Num paralelo às considerações de R. Brooks em [Bro91c, Bro91b], Etzioni desenvolve alguns agentes utilizando-se do

sistema de comandos no UNIX. Por exemplo, um agente efetor é formado por comandos transmitidos para o ambiente externo, a fim de mudar um estado. Exemplos são os comandos “*mv*” ou o “*compress*”. Um agente sensor é montado a partir de comandos que capturam informações do mundo externo, por exemplo, “*pwd*”, “*ls*”, “*finger*”, “*whois*”, etc. Finalizando, AS’s é uma área com muitas perguntas sem respostas e outras tantas ainda não formuladas [Mae95].

Apêndice A

A.1 Bases de Regras

Neste apêndice estão definidos os termos nebulosos das variáveis que compõem a camada sensorial. A camada sensorial é modelada por uma rede de agentes situados, segundo uma topologia fixa, a figura desta rede encontra-se no capítulo 5. O agente é composto por uma combinação de entradas e saídas dessas variáveis nebulosas.

DISTANCIA

X0	0	1	10	29
X1	19	35	37	57
X2	47	55	57	89
X3	67	98	99	100

ANDAR

A0	0	1	5	20
A1	10	25	27	40
A2	37	65	67	88
A3	80	89	99	100

INIBIR

I0	0	1	10	15
I1	8	29	30	40
I2	29	43	45	65
I3	60	65	98	100

RECURSOS

R0	0	5	10	15
R1	10	27	29	40
R2	30	49	51	70
R3	55	75	90	100

MOTOR

M0	0	1	2	15
M1	10	29	30	40
M2	30	55	60	65
M3	55	95	97	100

Base de regras para os agentes #1 e #3. Conjunto de regras com uma entrada e uma saída, ou seja, $X \rightarrow A$:

X0 A0
 X1 A0
 X1 A1
 X2 A1
 X2 A2
 X3 A3

Conjunto de regras para o agente #2. Conjunto de regras com uma entrada e duas saídas, ou seja, $X \rightarrow A \wedge I$:

X0 A0
 X1 A0
 X1 A1
 X2 A1
 X2 A2
 X3 A2
 X3 A3

e

X0 I0
 X0 I1
 X1 I1
 X1 I2
 X2 I2
 X2 I3
 X3 I3

Base de regras para os agentes #4 e #6. Conjunto de regras com duas entradas e duas saídas, ou seja, $A \wedge I \rightarrow R \wedge I$:

A0 I0 R1
 A0 I1 R0
 A1 I0 R1

A1 I1 R0
 A1 I1 R3
 A2 I0 R2
 A2 I1 R1
 A2 I2 R1
 A2 I0 R3

e

A0 I1 I0
 A0 I0 I0
 A1 I0 I1
 A1 I1 I2
 A1 I2 I1
 A2 I0 I2
 A2 I1 I2
 A2 I2 I0
 A2 I3 I0

Base de regras para o agente #5. Conjunto de regras com uma entrada e uma saída, ou seja, $A \rightarrow R$:

A0 R0
 A0 R1
 A1 R1
 A2 R2
 A2 R3
 A3 R2
 A3 R3

Base de regras do agente #8. Conjunto de regras com três entradas e uma saída, ou seja, $R \wedge I_i \wedge I_j \rightarrow M$:

R0 I0 I0 M0
 R0 I1 I1 M0
 R0 I0 I3 M0
 R0 I2 I0 M0
 R0 I2 I1 M0
 R0 I3 I0 M0
 R0 I1 I2 M0
 R0 I0 I1 M0
 R1 I0 I0 M1
 R1 I1 I1 M1

R1	I1	I0	M1
R1	I0	I3	M1
R1	I0	I1	M1
R1	I2	I3	M0
R1	I3	I2	M0
R1	I2	I2	M0
R2	I0	I0	M3
R2	I1	I0	M2
R2	I1	I1	M2
R2	I1	I2	M1
R2	I2	I1	M1
R2	I0	I3	M1
R3	I0	I0	M3
R3	I1	I0	M3
R3	I0	I1	M3
R3	I0	I3	M2
R3	I3	I0	M2
R3	I1	I2	M2
R3	I2	I1	M2

Base de regras dos agentes #7 e #9. Conjunto de regras com uma entrada e uma saída, ou seja, $R \rightarrow M$:

R0	M0
R0	M1
R1	M0
R1	M1
R2	M1
R2	M2
R3	M3

Referências Bibliográficas

- [AC91] Philip E. Agre and David Chapman. What are plans for. In Pattie Maes, editor, *Designing Autonomous Agents - Theory and Practice from Biology to Enginnering and Back*, Special Issues of Robotics and Autonomous Systems, pages 17–34. Bradford Book - MIT Press, 1991.
- [AD91] Tracy L. Anderson and Max Donath. Animal behavior as a paradigm for developing robot autonomy. In Pattie Maes, editor, *Designing Autonomous Agents - Theory and Practice from Biology to Enginnering and Back*, Special Issues of Robotics and Autonomous Systems, pages 145–168. Bradford Book - MIT Press, 1991.
- [Adl86] Klaus-Peter Adlassing. Fuzzi set theory in medical diagnosis. *IEEE Transactions on System, Man, and Cybernetics*, 16(2):260–265, apr 1986.
- [Agr95] Phillip E. Agre. Computational research on interaction and agency. *Artificial Intelligence*, 72:1–52, jan 1995.
- [AJ95] K. Shankar A.V. Joshi, S.C. Sahasrabudhe. Sensitivity of combination schemes under conflicting conditions and a new method. In Jacques Wainer and Ariadne Carvalho (Volume Editors), editors, *Advances in Artificial Intelligence*, number 991 in Lecture Notes in Computer Science, pages 39–48. Springer-Verlag, oct 1995. 12th. Brazilian Symposium on Artificial Intelligence SBIA'95, Campinas, Brazil.
- [Arb69] Michael A. Arbib. *Theories of Abstract Automata*. Prentice-Hall Series in Automatic Computation. Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1969.
- [Ark91] Ronald C. Arkin. Integrating behavioral, perceptual, and world knowledge in reactive navigation. In Pattie Maes, editor, *Designing Autonomous Agents - Theory and Practice from Biology to Enginnering and Back*, Special Issues of Robotics and Autonomous Systems, pages 105–122. Bradford Book - MIT Press, 1991.
- [Ark93] Ronald C. Arkin. Modeling neural function at the schema level: Implications and results for robotic control. In Roy E. Ritzmann Randall D. Beer
-

- and Thomas Mackenna, editors, *Biological Neural Networks in Invertebrate, Neuroethology and Robotics*, pages 383–410. Academic Press Inc., 1993. Part IV Robotics.
- [Aze96] M. Roisenberg; J.M. Barreto; F.M. Azevedo. Biological inspirations in neural networks implementations of autonomous agents. In *Advances in Artificial Intelligence*, Lecture Notes in Computer Science, pages 211–220. Springer-Verlag, oct 1996. 13th. Brazilian Symposium on Artificial Intelligence SBIA'96, Curitiba, Brazil.
- [Baa93] Nil A. Baas. Emergence, hierarchies, and hyperstructures. In Christopher G. Langton, editor, *Artificial Life III*, volume XVII of *Santa Fe Institute Studies in the Sciences of Complexity*, pages 515–537, Santa Fe, New Mexico, USA, 1993. Addison-Wesley Publishing Company. Proceedings of Third Artificial Life Workshop, in June of 1992.
- [BD94] Stefan Bussmann and Yves Demazeau. An agent model combining reactive and cognitive capabilities. In *IEEE International Conference on Intelligence Robots and Systems*, pages 1–8, Munich, sep 1994. IROS 94. It directly received from the authors.
- [BeM94] Y. Zhao; C.V. Ravishankar; S.L. BeMent. Coping with limited on-board memory and communication bandwidth in mobile-robot systems. *IEEE Transactions on Systems, Man and Cybernetics*, 24(1):58–72, 1994.
- [BF81] A. Barr and E.A. Feigenbaum. *The Handbook of Artificial Intelligence*, volume 1. William Kaufmann Inc., 1981. Los Altos, California.
- [Bit96] Guilherme Bittencourt. *Inteligência Artificial - Ferramentas e Teorias*. 10a. Escola de Computação. Editora da Unicamp, So Paulo, 1996.
- [Bit97] Guilherme Bittencourt. In the quest of the missing link. Personal Communication, 1997.
- [Bra94] Eugénio Oliveira; Helder Coelho; Pavel Brazdil. Artificial intelligence for multi-agent systems: Methodologies and applications. In Eugénio Oliveira, editor, *Advanced School on Artificial Intelligence '94*, volume I–II. Troia - Setubal - Portugal, sep 1994.
- [Bro86] Rodney A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, RA-2:14–23, mar 1986.
- [Bro90] Colin M. Angle; Rodney A. Brooks. Small planetary rovers. In *IEEE/RSJ International Workshop on Intelligent Robots and Systems*, pages 383–388, 1990. Ikaraba, Japan.
-

-
- [Bro91a] Rodney A. Brooks. Elephants don't play chess. In Pattie Maes, editor, *Designing Autonomous Agents - Theory and Practice from Biology to Engineering and Back*, Special Issues of Robotics and Autonomous Systems, pages 3–15. Bradford Book - MIT Press, 1991.
- [Bro91b] Rodney A. Brooks. Intelligence without reason. Technical Report MIT AI Lab Memo 1293, Massachusetts Institute of Technology - Artificial Intelligence Laboratory, apr 1991. Prepared for *Computers and Thought*, IJCAI-91, <http://www.ai.mit.edu/people/brooks/papers.html>.
- [Bro91c] Rodney A. Brooks. Intelligence without representation. *Artificial Intelligence*, 47:139–159, aug 1991.
- [Bro91d] Rodney A. Brooks. New approaches to robotics. *Science*, 253:1227–1232, sep 1991.
- [Bro92] Rodney Brooks. Artificial life and real robots. In Francisco J. Varela and Paul Bourguine, editors, *Toward a Practice of Autonomous Systems: Proc. of First European Conference on Artificial Life*. MIT Press, 1992. Cambridge, MA, USA.
- [Bro93] Rodney A. Brooks. A robot that walks; emergent behaviors from a carefully evolved network. In Roy E. Ritzmann Randall D. Beer and Thomas Mackenna, editors, *Biological Neural Networks in Invertebrate, Neuroethology and Robotics*, pages 355–363. Academic Press Inc., 1993. Part IV Robotics.
- [Car92] Gail Carpenter. Neural network model for pattern recognition and associative memory. In L. Nadel and D. Stein, editors, *1991 - Lectures in Complex Systems*, volume IV of *SFI Studies in the Sciences of Complexity*, pages 3–41. Addison-Wesley, 1992.
- [Car95] Norman Carver. Examining some assumptions of the fa/c distributed problem-solving paradigm. In *Proceedings of the Midwest Artificial Intelligence and Cognitive Science Society Conference*, apr 1995. <http://www.cs.siu.edu/carver/umass/fac-analysis-umass.html>.
- [CD93] Christopher M. Comer and John P. Dowd. Multisensory processing for movement: Antennal and cercal mediation of escape turning in the cockroach. In Roy E. Ritzmann Randall D. Beer and Thomas Mackenna, editors, *Biological Neural Networks in Invertebrate, Neuroethology and Robotics*, pages 89–112. Academic Press Inc., 1993. Part II Neuroethology II: Control of Orientation.
- [Cd94] B. Chaib-draa. Distributed artificial intelligence: An overview. In A. Kent and J. Williams, editors, *Encyclopedia of Computer Science and*
-

- Technology*, volume 31 (suppl. 16), pages 215–243. Marcel Dekker, INC, 1994. <http://iris.ift.ulaval.ca/publications/chaib/pub-95/pub-95.html>.
- [Cd95] B. Chaib-draa. Industrial applications in distributed ai. *Communications of the ACM*, 38(11):49–53, nov 1995. <http://iris.ift.ulaval.ca/publications/chaib/pub-95/pub-95.html>.
- [Cd96] Bernard Moulin; Brahim Chaib-draa. An overview of distributed artificial intelligence. In G.M.P. O'Hare and N.R. Jennings, editors, *Foundations of Distributed Artificial Intelligence*, chapter 1, pages 3–55. John Wiley & Sons. Inc, 1996. <http://iris.ift.ulaval.ca/publications/moulin/pub-96/pub-96.html>.
- [Chu74] K.L. Chung. *Elementary Probability Theory with Stochastic Processes*. Springer-Verlag, 1974. Apud in [Ste88], New York.
- [CJ93] Roy M. Turner; D.Richard Blidberg; Steven G. Chappell and James C. Jalbert. Generic behaviors: An approach to modularity in intelligent systems control. In *Proceedings of the 8th International Symposium on Unmanned Untethered Submersible Technology*, 1993. UNH CS TR #94-10.
- [CL73] C.K. Chang and R.C. Lee. *Symbolic Logic and Mechanical Theorem Proving*. Academic Press - New York, 1973.
- [CL95] Norman Carver; Zarko Cvetanovic and Victor Lesser. Sophisticated cooperation in fa/c distributed problem solving system. In *Proceedings of AAAI-91*, pages 191–198, 1995. <http://www.cs.siu.edu/carver/umass/fac-analysis-umass.html>.
- [Col95] Marco Dorigo; Marco Colombetti. Robot shaping: developing autonomous agents through learning. *Artificial Intelligence*, 72:321–370, jan 1995.
- [Cor81] V.L. Lesser; D.D. Corkill. Functionally accurate, cooperative distributed systems. *IEEE Transactions on Systems, Man and Cybernetics*, 11(1):81–96, jan 1981.
- [Cor89] E.H. Durfee; V.R. Lesser; D.D. Corkill. Trends in cooperative distributed problem solving. *IEEE Trans. Knowl. Data Eng. - KOE*, 11(1):63–83, 1989. Apud in [Cd96].
- [Cri89] C. Crick, F.; Asanuma. Certain aspects of the anatomy and physiology of cerebral cortex. In *Parallel Distributed Processing Explorations in the Microstructure of Cognition*, chapter Psychological and Biological Models, pages 333–371. MIT Press, 1989. Vol.II.
-

- [DB92] Sabine Berthet; Yves Demazeau and Olivier Boissier. Knowing each other better. In Glen Arbor, editor, 11th. *International Workshop on Distributed Artificial Intelligence*, pages 1–20, feb 1992. It directly received from the authors.
- [Dem93] Yves Demazeau. Le plate-forme paco et ses applications. *2ème Journée Nationale du PRC-IA sur les Systèmes Multi-Agents*, dec 1993. PRC-IA, Montpellier.
- [DM90] Yves Demazeau and Jean-Pierre Müller. Decentralized artificial intelligence. In Yves Demazeau and Jean-Pierre Müller, editors, *Decentralized AI*, volume 1, pages 3–13, 1990. Elsevier Science Publihers B.V.
- [DM91] Yves Demazeau and Jean-Pierre Müller. From reactive to intentional agents. In Yves Demazeau and Jean-Pierre Müller, editors, *Decentralized AI*, volume 2, pages 3–10, 1991. Elsevier Science Publihers B.V.
- [dO95] Pedro Paulo Balbi de Oliveira. *An Empirical Exploration of Computations of a Cellular-Automata-Based Artificial Life World*. PhD thesis, University of Sussex at Brighton, UK, 1995.
- [dP66] Luiz Miller de Paiva. *Medicina Psicossomática (Psicopatologia e Terapêutica)*. Livraria Editora Artes Médicas - São Paulo, 1966. 2a. edição.
- [DR94] Edmund H. Durfee and Jeffrey S. Rosenschein. Distributed problem solving and multi-agent systems: Comparisons and examples. In *Proceedings of the Thirteenth International Distributed Artificial Intelligence Workshop*, pages 94–104, jul 1994. <http://ai.eecs.umich.edu/diag/homepage.html>.
- [dS96] Claudio Cesar de Sá. Agentes autônomos. Monografia apresentada no Exame de Qualificação, no IEC/ITA, mar 1996.
- [dSGB096] Claudio Cesar de Sá; Guilherme Bittencourt and Nizam Omar. Architecture for a mobile autonomous agent. In Erol Gelenbe and Nestor Schmajuk, editors, *Biologically Inspired Autonomous Systems: Computation, Cognition, and Action Workshop*, Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708-0291, USA, mar 1996.
- [Dub79] H. Dubois, D.; Prade. Outline of fuzzy set theory: And introduction. In R.K. Ragade M.M. Gupta and R.R. Yager, editors, *Advances in Fuzzy Set Theory and Applications*, pages 27–48. North-Holland P.C., 1979.
-

-
- [Dur91] Edmund H. Durfee. The distributed artificial intelligence melting pot. *IEEE Transactions on Systems, Man and Cybernetics*, 21(6):1301–1306, nov 1991. Special Issue on Distributed Artificial Intelligence.
- [Etz93] Oren Etzioni. Intelligence without robots (a reply to brooks). *AI Magazine*, obtained by ftp in the Internet (etzioni@cs.washington.edu), dec 1993.
- [Far90] J. Doyne Farmer. A rosetta stone for connectionism. *Physica*, D-42:153–187, 1990. North-Holland.
- [Fer92] I.A. Ferguson. *TouringMachines: An Architecture for Dynamic, Rational, Mobile Agents*. PhD thesis, Clare Hall, University of Cambridge, UK, 1992. Apud in [WJ94].
- [FF85] G. Furbach, U.; Dirlich and C. Freksa. Towards a theory of knowledge representation systems. In W. Bibel and B. Petkoff, editors, *Artificial Intelligence: Methodology, Systems and Applications*, pages 77–84. Elsevier Science Publishers B.V., North-Holland, 1985. ECCAI.
- [FG94] Ricardo Gudwin Fernando Gomide. Controle de processos por lógica fuzzy. *Automação & Conrole*, pages 39–48, may 1994. Parte I.
- [Fil87] R. Tori; R. Arakaki; A.M.A. Massola; L.V.L. Filgueiras. *Fundamentos de Computação Gráfica - Compugrafia*. Engenharia da Computação. LTC - Livros Técnicos e Científicos Editora S.A., 1987. Páginas: 214–215.
- [FK94] Bernd Freisleben and Thomas Kunkelmann. Fuzzy/neural drive control of an autonomous vehicle (extended abstract). In *EULITE-Foundation, EUFIT '94*, pages 458–463, 1994. Promenade 9,D-52076 Aschen.
- [FL94] Haruo Takeda; Caludio Facchinetti and Jean-Claude Latombe. Planning the motions of a mobile robot in a sensory uncertainty field. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(10):1002–1017, oct 1994.
- [Fly89] Rodney A. Brooks; Anita M. Flynn. Fast, cheap and out of control: A robot invasion of the solar system. *Journal of The British Interplanetary Society*, 42:478–485, 1989.
- [FN71] R.E. Fikes and N.J. Nilsson. Strips: a new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2(3):189–208, 1971. Apud in [Agr95].
- [For90] Stephanie Forrest. Emergent computation: Self-organizing, collective, and cooperative phenomena in natural and artificial computing networks. *Physica*, D-42:1–11, 1990. North-Holland.
-

-
- [Gas89] M.N. Gasser, L.; Huhns. Themes in distributed artificial intelligence research. In Les Gasser and Michael N. Huhns, editors, *Distributed Artificial Intelligence*, pages 3–36. Pitman Publishing, 1989. Vol.II.
- [Gas91] Les Gasser. Social conceptions of knowledge and action: Dai foundations and opens systems semantics. *Artificial Intelligence*, 47:107–138, 1991.
- [GC86] Mario Goldsztein and Raúl Carnota. *Inteligencia Artificial Aplicada & Logica y Representation del Conocimiento*. I Escola Brasileiro-Argentina de Informática. Editora da Unicamp, fev 1986.
- [Gir88] J.P. Laumond; T. Simeon; R. Chatila; G. Giralt. Trajectory planning and motion control for mobile robots. In J.D. Boissonmat and J.P. Laumond, editors, *Geometry and Robotics*, number 391 in Lecture Notes in Computer Sciences, pages 132–149. Edited by G. Goos and J. Hartmanis, Springer Verlag, may 1988. Workshop in Toulouse - France.
- [GMP60] Eugene Galanter George Miller and Karl Pribram. *Plans and the Structure of Behavior*. Holt, New York, 1960. Apud in [Agr95].
- [GR95] Marco Oliveira; Mauricio Figueredo; Fernando Gomide and Luis Romero. Neurofuzzy navigation control and neural group selection. In *VI IFSA World Congress*, volume I, pages 73–76, 1995. São Paulo, Brazil.
- [Guy73] Arthur C. Guyton. *Tratado de Fisiologia Médica*. Guanabara Koogan, 1973. 4a. edição.
- [Hew81] W.A. Kornfeld; C. Hewitt. The scientific community metaphor. *IEEE Transactions on Systems, Man and Cybernetics*, 11(1):24–33, jan 1981.
- [Hil96] Melanie Hilario. An overview of strategies for neurosymbolic integration. It received from eletronic mail, hilario@cui.unige.ch, agu 1996.
- [Hog91] B.A. Hogg, T.; Huberman. Controlling chaos in distributed systems. *IEEE Transactions on Systems, Man and Cybernetics*, 21(6):1325–1332, nov 1991.
- [HP95] Frank Hoffmann and Gerd Pfister. A new learning method for the design of hierarchical fuzzy controllers using messy genetic algorithms. In *VI IFSA World Congress*, volume I, pages 249–252, 1995. São Paulo, Brazil.
- [HR95] Barbara Hayes-Roth. An architecture for adaptive intelligent systems. *Artificial Intelligence*, 72:329–365, jan 1995.
- [HW93] Ashley Walker; John Hallam and David Willshaw. Bee-havior in a mobile robot: The construction of a self-organized cognitive map and its use in robot navigation within a complex, natural environment. In *IEEE*
-

- International Conference on Neural Networks*, volume 3, pages 1451–1456, mar 1993. San Francisco, CA.
- [Ish95] Toru Ishida. Parallel, distributed and multi-agent production systems: A research. In *Foundation for Distributed Artificial Intelligence in International Conference on Multi-Agent Systems (ICMAS-95)*, pages 416–422, 1995. <http://www.lab7.kuis.kyoto-u.ac.jp/96/paper/index.html>.
- [Iwa93] T. Sawaragi; K. Itoh; O. Katai; S. Iwai. Integration of symbolic path-planning and fuzzy control for intelligent mobile robot. *Fuzzy Logic*, pages 579–587, 1993. Printed in the Netherlands.
- [Jen96] N. R. Jennings. Coordination techniques for distributed artificial intelligence. In G. M. P. O’Hare and N. R. Jennings, editors, *Foundations of Distributed Artificial Intelligence*, pages 187–210. Wiley, 1996. <http://www.elec.qmw.ac.uk/dai/publications/#1996>.
- [Kac87] J. Kacprzyk. Towards “human-consistent” decision support systems through commonsense knowledge-based-decision making and control models: A fuzzy logic approach. *Computer and Artificial Intelligence*, 2(6):97–122, 1987. Reprint in the Institute of Technical Cybernetics, Slovak Academy of Sciences, Czechoslovakia.
- [Kae95] Kenneth Basye; Thomas Dean; Leslie Pack Kaelbling. Learning dynamics: system identification for perceptually challenged agents. *Artificial Intelligence*, 72:139–171, jan 1995.
- [Kak96] Subhash C. Kak. Can we define levels of artificial intelligence? *Journal of Intelligent Systems*, 6, 1996. Obtained by ftp from: kak@ee.lsu.edu.us.
- [KH87] László T. Kóczy and Kaoru Hirota. A fast algorithm for fuzzy inference by compact rules. In R.K. Ragade M.M. Gupta and R.R. Yager, editors, *Fuzzy Logic for the Management of Uncertainty*, chapter 15, pages 297–317. North-Holland P.C., 1987.
- [KK68] Willian Kneale and Martha Kneale. *O Desenvolvimento da Lógica*. Fundação Calouste Gulbenkian, Av. de Berna, Lisboa, Portugal, 1968. Do original em inglês: “*The Development of Logic*”, Press Oxford.
- [Kle91] Mark Klein. Supporting conflict resolution in cooperative design systems. *IEEE Transactions on Systems, Man and Cybernetics*, 21(6):1379–1390, nov 1991.
- [Kli88] T. A. Klir, F.J.; Folger. *Fuzzy Sets, Uncertainty, and Information*. Prentice-Hall Int. Editions, 1988.
-

-
- [Koh84] T. Kohonen. *Self-Organization and Associative Memory*. Springer-Verlag, 1984. ISBN 3-540-12165-X.
- [Kos92] Seong-Gong Kong; Bart Kosko. Adaptative fuzzy systems for backing up a truck-and-trailer. *IEEE Transactions on Neural Networks*, 3(2):211–223, 1992.
- [Kos94] Bart Kosko. *Fuzzy Thinking: The New Science of Fuzzy Logic*. Flamingo, 1994. An Imprint of HarperCollins Manufacturing, Glasgow, UK.
- [Koz90] John R. Koza. Evolution and co-evolution of computer programs to control independently-acting agents. In *In Proceedings First Int. Conf. on Simulation of Adaptive Behavior, Paris*, pages 363–375. MIT Press, Cambridge, MA, 1990. Apud in [Bro92].
- [Koz91] John R. Koza. Evolving emergent wall following robotic behavior using the genetic programming paradigm. In *ECAL, Paris*, dec 1991. Apud in [Bro92].
- [Lan92] Christopher G. Langton. Artificial life. In L. Nadel and D. Stein, editors, *Lectures in Complex Systems*, volume IV of *SFI Studies in the Sciences of Complexity*, pages 189–241. Addison-Wesley, 1992.
- [Lau90] Jean Louis Laurière. *Problem Solving and Artificial Intelligence*. Prentice-Hall, New York, 1990.
- [Lee90a] Chuen Chien Lee. Fuzzy logic in control systems: Fuzzy logic controller, part i. *IEEE Transactions on Systems, Man and Cybernetics*, 20(2):404–418, mar 1990.
- [Lee90b] Chuen Chien Lee. Fuzzy logic in control systems: Fuzzy logic controller, part ii. *IEEE Transactions on Systems, Man and Cybernetics*, 20(2):419–435, apr 1990.
- [Les91] Susan E. Conry; Kazuhiro Kuwabara; Victor L. Lesser. Multistage negotiation for distributed constraint satisfaction. *IEEE Transactions on Systems, Man and Cybernetics*, 21(6):1462–1477, dec 1991.
- [Lip87] Richard Lippman. An introduction to computing with neural nets. *IEEE ASSP Magazine*, 37(3):4–22, apr 1987.
- [MA94] A.F. Rocha; F. Gomide; C. Mrooka and L. Alegre. Neurofuzzy systems in supervision and control, 1994. Dep. Computation and Atuomation, Faculty of Electrical Engineering, UNICAMP, Brazil.
- [MAA95] Jim-Shih Liaw Michael A. Arbib. Sensorimotor transformations in the world of frogs and robots. *Artificial Intelligence*, 72:53–79, jan 1995.
-

-
- [Mae91a] Pattie Maes, editor. *Designing Autonomous Agents - Theory and Practice from Biology to Engineering and Back*. Bradford Book - MIT Press, 1991.
- [Mae91b] Pattie Maes. Designing autonomous agents - theory and practice from biology to engineering and back. In *Designing Autonomous Agents - Theory and Practice from Biology to Engineering and Back*, Special Issues of Robotics and Autonomous Systems, pages 1–2. Bradford Book - MIT Press, 1991. Guest Editorial.
- [Mae91c] Pattie Maes. Situated agents can have goals. In Pattie Maes, editor, *Designing Autonomous Agents - Theory and Practice from Biology to Engineering and Back*, Special Issues of Robotics and Autonomous Systems, pages 49–70. Bradford Book - MIT Press, 1991.
- [Mae95] Pattie Maes. Intelligent software. *Scientific American*, pages 66–68, sep 1995.
- [Mam85] W.J.M. Kickert; E.H. Mamdani. Analysis of a fuzzy logic controller. *Fuzzy Sets and Systems*, 1(1):29–44, 1985.
- [Mar96] Maurício Marengoni. Reinforcement learning: Aprendizado por recompensa. University of Massachusetts, 215 Department of Computer Science, Amherst, MA 01003, USA, marengon@cs.umass.edu, 1996. Apostila do curso de férias no DEE/UFSC, em junho/96.
- [Mat92] Maja J. Matarić. Behavior-based control: Main properties and implications. In *IEEE International Conference on Robotics and Automation, Workshop on Architectures for Intelligent Control Systems*, pages 46–54, may 1992. Nice, France, <http://www.ai.mit.edu/people/maja/maja.html> (maja@cs.brandeis.edu).
- [Mat94] Maja J Matarić. Issues and approaches in design of collective autonomous agents. *Robotics and Autonomous Systems*, 16(2–4):321–331, dec 1994. <http://www.ai.mit.edu/people/maja/maja.html>.
- [Mat95] Maja J. Matarić. Evaluation of learning performance of situated embodied agents. In *Third European Conference on Artificial Life*, jun 1995. Granada, Spain, <http://www.ai.mit.edu/people/maja/maja.html> (maja@cs.brandeis.edu).
- [MB90] Pattie Maes and Rodney Brooks. Learning to coordinate behaviors. In *Conference on Artificial Intelligence (AAAI-90)*, Boston, MA, pages 796–802, 1990. It directly obtained from the author.
-

-
- [MB94] G. Meeden, L.; McGraw and D. Blank. Emergent control and planning in an autonomous vehicle. In *Fifteenth Annual Conference of the Cognitive Science Society*, nov 1994. pre-print.
- [McC95] John McCarthy. Making robots conscious of their mental states. <http://www-formal.stanford.edu/jmc/>, jul 1995.
- [Mer96] Ron Sun; Todd Peterson; Edward Merrill. A hybrid architecture for learning reactive sequential decision making. In *AAAI-96, Fall Symposium on Learning Complex Behavior*, 1996. MIT, Cambridge, MA.
- [Mey86] A. Meystel. Knowledge-based controller for intelligent mobile robots, 1986. Drexel University, Philadelphia, PA 19104, USA.
- [MF93] Melanie Mitchell and Stephanie Forrest. Genetic algorithms and artificial life. Technical Report TR. No. 93-11-072, Department of Computer Science, College of Engineering, UNM, University of New Mexico, Albuquerque, New Mexico 87131-1386, USA, 1993. Santa Fe Institute Working Paper 93-11-072, in Artificial Life.
- [Miz95] Masaharu Mizumoto. Are max and min suitable operations for fuzzy control? In *VI IFSA World Congress*, volume I, pages 497–500, 1995. São Paulo, Brazil.
- [MT92] David A. Cohn; Tom Mitchell and Sebastian Thrun. Robot learning. Summary of the post-NIPS workshop, dec 1992.
- [OD94] Michel Occello and Yves Demazeau. Building real time agents using parallel blackboards and its use for mobile robotics. In *IEEE International Conference on Systems, Man and Cybernetics*, volume II, pages 692–697, oct 1994. Texto da Advanced School on AI'94 – Portugal – sept/94.
- [PA94] Melanie Hilario; Christian Pellegrini and Frédéric Alexandre. Modular integration of connectionist and symbolic processing in knowledge-based systems. In *First Annual International Symposium on Integrating Knowledge and Neural Heuristics*, pages 123–132, may 1994.
- [Pay91] David W. Payton. Internalized plans: A representation for action resources. In Pattie Maes, editor, *Designing Autonomous Agents - Theory and Practice from Biology to Engineering and Back*, Special Issues of Robotics and Autonomous Systems, pages 89–103. Bradford Book - MIT Press, 1991.
- [Pen91] Roger Penrose. *A Mente Nova do Rei – Computadores, Mentes e as Leis da Física*. Editora Campus, 1991. Tradução de Waltesir Dutra.
-

-
- [Pet95] Ron Sun; Todd Peterson. A hybrid learning model for reactive sequential decision making. In *IJCAI'95 workshop on connectionist-symbolic integration*, 1995. Montreal, CA.
- [PM96] Ron Sun; Todd Peterson and Edward Merrill. Bottom-up skill learning in reactive sequential task. rsun@cs.ua.edu, apr 1996.
- [Pra94] Stephen Prata. *Divertindo-se com A Vida Artificial*. Berkeley Brasil, 1994. Tradução do original: Artificial Life Playhouse.
- [Rag79] M.M. Ragade, R.K.; Gupta. Fuzzy set theory and applications: A synthesis. In R.K. Ragade M.M. Gupta and R.R. Yager, editors, *Advances in Fuzzy Set Theory and Applications*, pages 19–25. North-Holland P.C., 1979.
- [RB96] Justinian P. Rosca and Dana H. Ballard. Evolution-based discovery of hierarchical behaviors. In *Conference on Artificial Intelligence (AAAI-96)*. AAAI Press / The MIT Press, 1996. obtained by ftp in the Internet (rosca,dana@cs.rochester.edu).
- [RDBS91] Hillel J. Chiel Randall D. Beer and Leon S. Sterling. A biological perspective on agent autonomous design. In Pattie Maes, editor, *Designing Autonomous Agents - Theory and Practice from Biology to Enginnering and Back*, Special Issues of Robotics and Autonomous Systems, pages 169–186. Bradford Book - MIT Press, 1991.
- [Ric88] Bradley L. Richards. When facts get fuzzy. *BYTE*, pages 285–290, apr 1988.
- [RK94] Elaine Rich and Kevin Knight. *Inteligência Artificial*. Makron do Brasil Editora Ltda. (MacGraw-Hill), So Paulo, 1994. 2a. edição.
- [RM93] Randall D. Beer; Roy E. Ritzmann and Thomas Mackenna. *Biological Neural Networks in Invertebrate, Neuroethology and Robotics*. Academic Press Inc., 1993. Papers originally presented at Workshops Held at the Academic of Sciences Study Center in Wood Hole - Ma, USA, in the fall of 1991.
- [Roc92] A.F. Rocha. *Neural Nets: A Theory for Brains and Machines*, volume 638 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, 1992.
- [Ros94] Edmund H. Durfee; Piotr J. Gmytrasiewicz; Jeffrey S. Rosenschein. The utility of embedded communications: Toward the emergence of protocols. In *In Proceedings of the Thirteenth International Distributed Artificial Intelligence Workshop*, pages 85–93, jul 1994. <http://ai.eecs.umich.edu/diag/homepage.html>.
-

-
- [Sej89] T.J. Sejnowski. Open questions about computation in cerebral cortex. In *Parallel Distributed Processing Explorations in the Microstructure of Cognition*, chapter Psychological and Biological Models, pages 372–389. MIT Press, 1989. Vol.II.
- [Slo94] Steven Sloman. The psychology of associative and symbolic reasoning. In *First Annual International Symposium on Integrating Knowledge and Neural Heuristics*, pages 227–236, may 1994.
- [Slo96a] Aaron Sloman. A systems approach to consciousness (how to avoid talking nonsense?). Summary of lecture presented at the Royal Society of Arts, London, UK, feb 1996.
- [Slo96b] Aaron Sloman. What sort of architecture is required for a human-like agent? In *Cognitive Modeling Workshop, AAAI96*, aug 1996. Portland, Oregon, USA – <http://www.cs.bham.ac.uk/axs>.
- [Smi83] R. Davis; R.G. Smith. Negotiation as a metaphor distributed solving problem. *Artificial Intelligence*, 20(1):63–109, 1983.
- [SS96] Michael Stiber and José P. Segundo. Learning in neural models with complex dynamics. obtained by ftp in the Internet (stiber@cs.ust.hk) and (iaqfjps@mvs.oac.ucla.edu), jul 1996.
- [Ste88] Luc Steels. Cooperation between distributed agents through self-organisation. *Journal on Robotics and Autonomous Systems*, nov 1988. This paper was presented at the Lagos Workshop on Autonomous Agents. North-Holland, Amsterdam.
- [Ste91] Luc Steels. Exploiting analogical representations. In Pattie Maes, editor, *Designing Autonomous Agents - Theory and Practice from Biology to Enginnering and Back*, Special Issues of Robotics and Autonomous Systems, pages 71–88. Bradford Book - MIT Press, 1991.
- [Ste94a] Luc Steels. The alife approach to ai, 1994. Lectures Notes of Course on Artificial Life, in the IX Computing School - Recife - PE - Brazil.
- [Ste94b] Luc Steels. The artificial life roots of artificial intelligence. *Artificial Life Journal*, 1(1), 1994. MIT Press, Cambridge.
- [Ste94c] Luc Steels. Emergent functionality in robotic agents through on-line evolution. 1994. MIT Press, Cambridge.
- [Ste94d] Luc Steels. Mathematical analysis of behavior systems. In *Proceedings of the Perarc Conference, Lausanne.*, 1994. <http://arti.vub.ac.be/www/memos/publications.html#steels>.
-

-
- [Ste94e] Rodney A. Brooks; Lynn Andrea Stein. Building brain for bodies. *Autonomous Robots*, 1:7–25, 1994.
- [Ste95a] Luc Steels. Intelligence - dynamics and representations. *The Biology and Technology of Intelligent Autonomous Agents*, 1995. <http://arti.vub.ac.be/www/steels/publications.html>.
- [Ste95b] Luc Steels. When are robots intelligent autonomous agents? *Journal of Robotics and Autonomous Systems*, 15:3–9, 1995. <http://arti.vub.ac.be/www/steels/publications.html>.
- [Sug85] M. Sugeno, M; Nishida. Fuzzy control of model car. *Fuzzy Sets and Systems*, 6(2):103–113, 1985.
- [SV93] N.N. Sulzberger, S.M.; Tschichold-Gürman and S.J. Vestli. Fun: Optimization of fuzzy rule based systems using neural networks. In *IEEE International Conference on Neural Networks*, volume I, pages 312–316, mar 1993. San Francisco, CA, USA.
- [SW95] Alessandro Saffiotti and Leonard P. Wesley. Hierarchical fuzzy-based localization in autonomous mobile robots. In *VI IFSA World Congress*, volume I, pages 141–144, 1995. São Paulo, Brazil.
- [TG96] Christian W. Omlin; Karvel K. Thornber and C. Lee Giles. Fuzzy finite-state automata can be deterministically encoded into recurrent neural networks. Technical Report Technical Report CS-TR-3599 and UMIACS-96-12, University of Maryland, NEC Research Institute, Princeton, NJ 08540, USA and UMIACS, U. of Maryland, College Park, MD 20742, USA, apr 1996. <http://www.neci.nj.nec.com/homepages/giles.html>, <http://www.neci.nj.nec.com/homepages/omlin/omlin.html>, <http://www.cs.umd.edu/TRs/TR-no-abs.html>.
- [Thr94] Sebastian Thrun. A lifelong learning perspective for mobile robot control. In *Conference on Intelligent Robots and Systems*, Munich, Germany, sep 1994. Proceedings of IEEE.
- [Tor96a] Kari Torkkola. Blind separation of convolved sources based on information maximization. In *IEEE Workshop on Neural Networks for Signal Processing*. Kyoto, Japan, sep 1996. obtained by ftp in the Internet (A540AA@email.mot.com).
- [Tor96b] Kari Torkkola. Blind separation of delayed sources based on information maximization. In *IEEE International Conference on Acoustics, Speech & Signal Processing*, Atlanta, GA, USA, may 1996. obtained by ftp in the Internet (A540AA@email.mot.com).
-

- [Tou88] Pierre Tournassoud. Motion planning for mobile robot with a kinematic constraint. In J.D. Boissonmat and J.P. Laumond, editors, *Geometry and Robotics*, number 391 in Lecture Notes in Computer Sciences, pages 150–171. Edited by G. Goos and J. Hartmanis, Springer Verlag, may 1988. Workshop in Toulouse - France.
- [TS87] K. Terano, T.; Asai and M. Sugeno. *Fuzzy Systems Theory and Its Applications*. Academic Press INC, 1987. London.
- [Tsc94] Nadine Tschichold-Gürman. Fuzzy rulenet: An artificial neural networks for fuzzy classification. In *1994 Symposium on Applied Computing (SAC '94) - ACM Computer Science Conference*, mar 1994. This paper was directly received from the author. Phoenix, USA.
- [Tur91] Roy M. Turner. The tragedy of the commons and distributed ai systems. Technical Report 93-01, University of New Hampshire, Department of Computer Science, jan 1991. In *Proceedings of the 12th International Workshop on Distributed Artificial Intelligence*, Hidden Valley, PA.
- [Tur93] Roy M. Turner. Context-sensitive reasoning for autonomous agents and cooperative distributed problem solving. In *Proceedings of the 1993 IJCAI Workshop on Using Knowledge in Its Context*, 1993. Chambúry, France, UNH CS TR #94-09, <http://cdps.umcs.maine.edu/Papers/papers.html>.
- [Tur95] Roy M. Turner. Context-sensitive, adaptive reasoning for intelligent auv control: Orca project update. In *Proceedings of the 9th International Symposium on Unmanned Untethered Submersible Technology (AUV'95)*, 1995. Durham, New Hampshire, <http://cdps.umcs.maine.edu/Papers/papers.html>.
- [Vio93] Greg Viot. Fuzzy logic in *c. Dr. Dobb's Journal*, pages 41–50, feb 1993.
- [VO92] Jari Vaario and Stsuo Ohsuga. An emergent construction of adaptive neural architectures. *Heuristics - The Journal fo Knowledge Engineering*, page 19, jul 1992. Directly received from the author: jari@ohsuga.rcast.u-tokyo.ac.jp.
- [vP94a] U.R. Zimmer; C. Fischer; E. von Puttkamer. Navigation topologic feature-maps. In *IIZUKA '94*, aug 1994. Fukuoka, Japan.
- [vP94b] U.R. Zimmer; E. von Puttkamer. Comparing world-modelling strategies for autonomous mobile robots. In *IWK '94*, sep 1994. Ilmenau, Germany.
- [vP94c] U.R. Zimmer; E. von. Puttkamer. Realtime-learning on an autonomous mobile robot with neural networks. In *Euromicro '94*, jun 1994. Vaester-aas, Sweden.
-

-
- [Wak93] Hyroyoshi Nomura; Isao Hayashi; Noboru Wakami. A self-tuning of fuzzy inference rules by descent method. *Fuzzy Logic*, pages 465–475, 1993. Printed in the Netherlands.
- [Wel91] Michael P. Wellman. Book review - the ecology of computation. Elsevier - Artificial Intelligence, 52, pp: 205–218, 1991. Ed. B.A. Huberman.
- [Wil83] David Wilkins. Domain-independent planning: Representation and plan generation. Technical Report Technical Note No. 266R, SRI International, Artificial Intelligence Center, Computer Science and Technology Division, 333 Ravenswood Ave., Menlo Park, CA 94025, may 1983. The research reported herein was supported by the Air Force Office of Scientific Research, Contract F4920-79-C0188, SRI Project 8871.
- [Win87] Patrick Henry Winston. *Inteligência Artificial*. Addison-Wesley Publishing, Reading. 2 Edio, Massachusetts, 1987. LTC - Livros Técnicos e Científicos Editora S.A.
- [WJ94] M.J. Wooldridge and N.R. Jennings. Agent theories, architectures, and languages: A survey. In M.J. Wooldridge and N.R. Jennings, editors, *Proc. ECAI-Workshop on Agent Theories, Architectures and Languages*, pages 1–32, 1994.
- [WK93] Edmund A. Arbas; Mark A. Willis and Ryohei Kanzaki. Organization of goal-oriented locomotion: Pheromone-modulated flight behavior of moths. In Roy E. Ritzmann Randall D. Beer and Thomas Mackenna, editors, *Biological Neural Networks in Invertebrate, Neuroethology and Robotics*, pages 159–198. Academic Press Inc., 1993. Part II Neuroethology II: Control of Orientation.
- [Yae93] Larry Yaeger. Computational genetics, physiology, metabolism, neural systems, learning, vision, and behavior or polyworld: Life in a context. In Christopher G. Langton, editor, *Artificial Life III*, volume XVII of *Santa Fe Institute Studies in the Sciences of Complexity*, pages 263–298, Santa Fe, New Mexico, USA, 1993. Addison-Wesley Publishing Company. Proceedings of Third Artificial Life Workshop, in June of 1992.
- [Yag84] Ronald R. Yager. Approximate reasoning as a basis for rule-based expert systems. *IEEE Transactions on Systems, Man and Cybernetics*, 14(4):636–643, jul–aug 1984.
- [Zad94] Lotfi A. Zadeh. Fuzzy logic, neural networks, and soft computing. *Communications of the ACM*, 37(3):77–84, mar 1994.
- [Zan90] R. Serra; G. Zanarini. *Complex Systems and Cognitive Processes*. Springer-Verlag, 1990. Heidelberg, Germany.
-

- [Zel88] A. Zelinsky. Environment mapping with a mobile robot using sonar. In C.J. Barker and M.J. Brook, editors, *Lecture Notes in AI*, number 406 in Subseries of Lecture Notes in CS, pages 363–378. J. Siekmann, nov 1988. Edited in 2nd. Australian Joint Artificial Intelligence, Adelaide, Australia.
- [Zie92] J.-Y. Antoine; O. Baujard; O. Boissier; B. Caillot; M. Chaillot; Yves Demazeau; S. Pesty; J. Sichman; M.-H. Stéfanini; D. Ziebelin. Vers une taxinomie du vocabulaire pour les systèmes multi-agents. In *Journée Nationale du PRC-IA sur les Systèmes Multi-Agents*, pages 1–12, Nancy, France, dec 1992. PRC-IA. It directly received from the authors, Pôle PLEIAD (Pôle et Lieu d 'Echanges en Intelligence Artificielle Distribuée in Grenoble).
- [Zim94] P. Lefèvre; A. Prüb; U.R. Zimmer. Alice-topographic exploration, cartography and adaptive navigation on a simple mobile robots. In *ISRPC '94*, jun 1994. Leeawennorst, Netherlands.
-