

POLICARPO BATISTA ULIANA

**CONTRIBUIÇÕES PARA A ÁREA DE
INTELIGÊNCIA ARTIFICIAL BASEADAS
EM UMA ABORDAGEM HOLÍSTICA**

**FLORIANÓPOLIS
2002**

UNIVERSIDADE FEDERAL DE SANTA CATARINA

**PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA ELÉTRICA**

**CONTRIBUIÇÕES PARA A ÁREA DE
INTELIGÊNCIA ARTIFICIAL BASEADAS
EM UMA ABORDAGEM HOLÍSTICA**

Tese submetida à
Universidade Federal de Santa Catarina
como parte dos requisitos para a
obtenção do grau de Doutor em Engenharia Elétrica.

POLICARPO BATISTA ULIANA

Florianópolis, Agosto de 2002.

CONTRIBUIÇÕES PARA A ÁREA DE INTELIGÊNCIA ARTIFICIAL BASEADAS EM UMA ABORDAGEM HOLÍSTICA

Policarpo Batista Uliana

‘Esta Tese foi julgada adequada para obtenção do Título de Doutor em Engenharia Elétrica, Área de Concentração em *Sistemas de Informação*, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Santa Catarina.’

Prof. Rui Seara, Dr.
Orientador

Prof. Edson Roberto De Pieri, Ph.D.
Coordenador do Programa de Pós-Graduação em Engenharia Elétrica

Banca Examinadora:

Prof. Rui Seara, Dr.
Presidente

AGRADECIMENTOS

Agradeço

aos que já não estão mais conosco:

Meu Pai - João

Meus avôs - Jorge, Analia e Abel

aos que vieram antes e nos mostraram o caminho:

Minha mãe - Salete

Meu orientador - Prof. Rui

aos que andaram ao nosso lado e nos ajudaram a crescer:

Meus sócios - Guilherme e Jurandir

Meus amigos - Adilson, Luis, Sid, Moacir e Surena

aos que vieram depois e nos amaram e inspiraram:

Minhas paixões - Karla, Marisa e Lissandra

Minha princesinha - Maria Eduarda

Resumo da Tese apresentada à UFSC como parte dos requisitos necessários para a obtenção do grau de Doutor em Engenharia Elétrica.

CONTRIBUIÇÕES PARA A ÁREA DE INTELIGÊNCIA ARTIFICIAL BASEADAS EM UMA ABORDAGEM HOLÍSTICA

Policarpo Batista Uliana

Agosto/2002

Orientador: Prof. Rui Seara, Dr.

Área de Concentração: Sistemas de Informação

Palavras-chave: inteligência artificial; agente cognitivo; esquemas de transição; memória holográfica; treinamento supervisionado.

Número de Páginas: 143

RESUMO: Na presente tese de doutorado três novas contribuições para a área de inteligência artificial são apresentadas. Elas foram desenvolvidas visando o aprimoramento de um “agente cognitivo” que “vive” em um mundo virtual. O agente deve ser capaz de aprender uma série de habilidades e comportamentos ditos “inteligentes”.

A primeira contribuição é uma metodologia para definição de escalas para a medição de parâmetros relacionados com o grau de “inteligência” de agentes cognitivos naturais e artificiais. Esse tipo de escala é de fundamental importância para o desenvolvimento de sistemas efetivamente inteligentes, pois possibilita a quantificação de certos tipos de “inteligência”.

A segunda e mais importante contribuição deste trabalho é um novo modelo de processamento baseado em estruturas denominadas “esquemas de transição”. Esse modelo quebra o paradigma tradicional de processamento de sinais, não fazendo distinção entre entradas e saídas. O modelo proposto permite a hierarquização das estruturas de controle de um agente cognitivo, que podem ser treinadas separadamente, viabilizando o aprendizado rápido de processos bastante complexos, que não são praticamente exequíveis com os modelos tradicionais.

Por fim, apresentamos um novo modelo de memória denominado memória associativa holográfica. Esse tipo de memória apresenta características comparáveis às da memória humana, podendo ser implementada através de redes neurais artificiais baseadas em padrões de pesos randômicos para armazenar a informação desejada.

Estes resultados são frutos de uma abordagem holística, na qual, além das áreas tradicionais de engenharia e computação, buscamos subsídios nas áreas de ciências humanas de psicologia e filosofia, através do modelo construtivista de Jean Piaget e do modelo semiótico de Charles Peirce.

Abstract of Thesis presented to UFSC as a partial fulfillment of the requirements for the degree of Doctor in Electrical Engineering.

CONTRIBUTIONS TO THE ARTIFICIAL INTELLIGENCE AREA BASED ON A HOLISTIC APPROACH

Policarpo Batista Uliana

August/2002

Advisor: Prof. Rui Seara, Dr.

Area of Concentration: Information Systems.

Keywords: artificial intelligence; cognitive agent; transition schemes; holographic memory; supervised training.

Number of Pages: 143

ABSTRACT: In this thesis, three contributions in the artificial intelligence area are introduced. They have been developed aiming at the refinement of a “cognitive agent” that “lives” in a virtual world. The agent should be able to learn a series of abilities and behaviors to be considered “intelligent”.

The first contribution is a methodology to define scales for the measurement of parameters related to the degree of natural and artificial intelligence of cognitive agents. Such a measurement type is of fundamental importance for the effective improvement of intelligent systems, since it permits to quantify certain classes of “intelligence”.

The second and the most important contribution of this thesis is the development of a processing model based on an approach termed “transition schemes”. This model breaks the traditional paradigm of the signal processing systems (input-system-output), which does not make distinction between inputs and outputs. The proposed model permits an hierarchy of the control structures of a cognitive agent, which can be separately trained making it possible to learn very quickly complex procedures; such procedures could not be effectively carried out by traditional models.

Finally, in the third part of this thesis, we have introduced a new memory model named holographic associative memory. This type of memory presents comparable characteristics to that of a human memory, which can be implemented through artificial neural networks using a data configuration based on random weight patterns to store the desired information.

The results obtained in this study are consequences of the use of a holistic approach, which, besides from the traditional areas of engineering and computation, have borrowed from other areas as psychology, philosophy and human sciences, by using both the Jean Piaget’s constructivist model and Charles Pierce’s semiotic model.

SUMÁRIO

LISTA DE FIGURAS	ix
LISTA DE TABELAS	x
1 INTRODUÇÃO	1
1.1 A BUSCA POR SISTEMAS “INTELIGENTES”.....	1
1.2 NECESSIDADE DE SISTEMAS “INTELIGENTES”.....	4
1.3 A PLAUSIBILIDADE DOS SISTEMAS “INTELIGENTES” ARTIFICIAIS.....	6
1.4 NECESSIDADE DE NOVOS PARADIGMAS DE IA.....	7
1.5 UM NOVO PARADIGMA DE CONTROLE E PROCESSAMENTO DE SINAIS.....	8
1.6 HISTÓRICO DO TRABALHO.....	10
1.7 ORGANIZAÇÃO DO TRABALHO.....	14
2 ABORDAGENS SEMIÓTICA E CONSTRUTIVISTA	16
2.1 INTRODUÇÃO.....	16
2.2 ABORDAGEM SEMIÓTICA.....	17
2.3 ABORDAGEM CONSTRUTIVISTA.....	18
2.4 AGENTES COGNITIVOS.....	20
2.5 ABORDAGEM PROPOSTA.....	22
3 ESCALAS DE MEDIÇÃO DOS GRAUS DE ADAPTABILIDADE E INTELIGÊNCIA DE AGENTES COGNITIVOS	24
3.1 INTRODUÇÃO.....	24
3.2 ESCALAS PROPOSTAS.....	26
3.3 MEDIÇÃO DO GRAU DE ADAPTABILIDADE.....	29
3.4 EXEMPLO DE ESCALA PARA MEDIÇÃO DO GRAU DE ADAPTABILIDADE.....	32
3.4.1 DESCRIÇÃO DA ESCALA.....	33
3.5 MEDIÇÃO DO GRAU DE INTELIGÊNCIA SIMBÓLICA.....	38
3.6 EXEMPLO DE ESCALA PARA MEDIÇÃO DO GRAU DE INTELIGÊNCIA SIMBÓLICA.....	40
3.6.1 DESCRIÇÃO DA ESCALA.....	41
3.7 CONSIDERAÇÕES SOBRE AS ESCALAS PROPOSTAS.....	44
3.8 RESULTADOS EXPERIMENTAIS.....	44
3.9 CONCLUSÕES.....	45
4 UM NOVO PARADIGMA DE CONTROLE BASEADO EM ESQUEMAS DE TRANSIÇÃO APLICADO À CRIAÇÃO DE AGENTES COGNITIVOS	47
4.1 INTRODUÇÃO.....	47
4.2 EXEMPLO DE PROCESSAMENTO DE OBJETIVOS.....	48
4.3 TREINAMENTO DE AGENTES AUTÔNOMOS.....	49
4.4 TREINAMENTO COM REFORÇO.....	50
4.5 MODELO PROPOSTO.....	53

4.6	MODELOS DE ESQUEMAS	54
4.7	ESQUEMAS DE TRANSIÇÃO.....	56
4.8	CLASSIFICAÇÃO DOS ESQUEMAS DE TRANSIÇÃO.....	57
4.9	METAESQUEMAS DE TRANSIÇÃO	59
4.10	CONTROLE DE UM METAESQUEMA DE TRANSIÇÃO.....	61
4.11	REPRESENTAÇÃO DE METAESQUEMAS DE TRANSIÇÃO.....	62
4.12	IMPLEMENTAÇÃO DE METAESQUEMAS DE TRANSIÇÃO.....	64
4.13	EXEMPLO DE IMPLEMENTAÇÃO DE METAESQUEMAS DE TRANSIÇÃO.....	65
4.14	EXEMPLOS DIDÁTICOS DE ESQUEMAS DE TRANSIÇÃO.....	65
4.15	DEFINIÇÃO DE METAESQUEMAS DE TRANSIÇÃO.....	74
4.16	MANUTENÇÃO DE METAESQUEMAS DE TRANSIÇÃO.....	74
4.17	MECANISMOS DE EXPLORAÇÃO.....	77
4.18	PROCESSAMENTO DE “OBJETIVOS”.....	79
4.19	PROCESSAMENTO BÁSICO DE METAESQUEMAS DE TRANSIÇÃO.....	80
4.20	TRATAMENTO DE OBJETIVOS MÚLTIPLOS.....	83
4.21	NEGAÇÃO DE OBJETIVOS	84
4.22	ESCOLHA DE CAMINHOS	85
4.23	DEFINIÇÃO DE OBJETIVOS DE SAÍDA.....	87
4.24	CONSIDERAÇÕES SOBRE ESTRUTURAS BASEADAS EM METAESQUEMAS DE TRANSIÇÃO.....	89
4.25	RESULTADOS EXPERIMENTAIS.....	90
4.26	CONTROLE DO BRAÇO DO ROBÔ.....	91
4.27	CONCLUSÕES.....	97
5	MEMÓRIAS ASSOCIATIVAS HOLOGRÁFICAS.....	100
5.1	INTRODUÇÃO.....	100
5.2	MEMÓRIAS ASSOCIATIVAS.....	101
5.3	MEMÓRIAS ASSOCIATIVAS ENCONTRADAS NA LITERATURA.....	102
5.3.1	MEMÓRIA ASSOCIATIVA LINEAR (LAM).....	102
5.3.2	MEMÓRIA ASSOCIATIVA DE SEGUNDA ORDEM (SAM).....	103
5.3.3	MEMÓRIA RECURSIVA AUTO-ASSOCIATIVA (RAAM).....	103
5.3.4	MEMÓRIA BASEADA EM CONVOLUÇÃO CIRCULAR (HRR).....	103
5.4	MEMÓRIA ASSOCIATIVA HOLOGRÁFICA.....	104
5.5	PRINCÍPIO DE OPERAÇÃO DA MEMÓRIA ASSOCIATIVA HOLOGRÁFICA.....	106
5.6	ROBUSTEZ E CAPACIDADE DE ARMAZENAMENTO DA MAH.....	108
5.7	APLICAÇÕES DA MEMÓRIA ASSOCIATIVA HOLOGRÁFICA.....	109
5.7.1	ARMAZENAMENTO DE SEQÜÊNCIAS.....	109
5.7.2	ARMAZENAMENTO DE CONJUNTO DE SEQÜÊNCIAS	110
5.7.3	REPRESENTAÇÃO DE VARIÁVEIS.....	110
5.7.4	REPRESENTAÇÃO DE PREDICADOS.....	110
5.7.5	REPRESENTAÇÃO DE QUADROS	111
5.8	RESULTADOS EXPERIMENTAIS.....	111
5.9	PLAUSIBILIDADE BIOLÓGICA.....	112

5.10 CONCLUSÕES.....	113
6 CONCLUSÕES.....	115
6.1 RESULTADOS OBTIDOS.....	116
6.2 CONSIDERAÇÕES FINAIS.....	118
APÊNDICE 1: EXEMPLOS DE APLICAÇÃO DE METAESQUEMAS DE TRANSIÇÃO.....	120
A.1 INTRODUÇÃO.....	120
A.2 PROBLEMA DA MÁQUINA CAÇA-NÍQUEIS.....	120
A.3 CONTROLE DE UM AGENTE QUE “COME” OBJETOS.....	123
A.4 CONTROLE DE UM “ELEVADOR”.....	126
A.5 PROBLEMA DO “JOGO DA VELHA”.....	128
APÊNDICE 2: CAPACIDADE DE ARMAZENAMENTO DE DADOS DA MEMÓRIA ASSOCIATIVA HOLOGRÁFICA.....	135
B.1 INTRODUÇÃO.....	135
B.2 DESCRIÇÃO.....	135
REFERÊNCIAS BIBLIOGRÁFICAS.....	138

LISTA DE FIGURAS

FIG. 1.	MAPEAMENTOS: (a) SEMIÓTICA E (b) INTELIGÊNCIA ARTIFICIAL.....	17
FIG. 2.	INTERAÇÃO ENTRE O AGENTE COGNITIVO E O MEIO AMBIENTE.....	21
FIG. 3.	EXEMPLOS DE OBJETOS.....	34
FIG. 4.	EXEMPLOS DE OBJETO DISTORCIDO POR RUÍDO.....	36
FIG. 5.	REPRESENTAÇÃO DE UM AGENTE COGNITIVO INSERIDO NO MEIO AMBIENTE.....	51
FIG. 6.	MODELO DE CONTROLE HIERÁRQUICO.....	53
FIG. 7.	REPRESENTAÇÃO DE UM ESQUEMA DE TRANSIÇÃO.....	57
FIG. 8.	ENTRADAS E SAÍDAS DE UM METAESQUEMA DE TRANSIÇÃO.....	60
FIG. 9.	MODELO DE CONTROLE TRADICIONAL OBTIDO A PARTIR DE UM METAESQUEMA DE TRANSIÇÃO.....	60
FIG. 10.	EXEMPLO DE HIERARQUIA DE METAESQUEMAS.....	61
FIG. 11.	INTERLIGAÇÃO DE UM METAESQUEMA DE TRANSIÇÃO.....	63
FIG. 12.	REPRESENTAÇÃO COMPLETA DE UM METAESQUEMA.....	67
FIG. 13.	INTERLIGAÇÃO DOS METAESQUEMAS PARA CONTROLE DE TEMPERATURA.....	68
FIG. 14.	REPRESENTAÇÃO GRÁFICA DO ESPAÇO DE EFEITO DADO PELA TABELA 8.....	81
FIG. 15.	TELA DO SIMULADOR DO AGENTE DEFINIDO NESTE PROBLEMA.....	91
FIG. 16.	SIMULADOR DO PROBLEMA DE CAÇA-NÍQUEIS, MOSTRANDO ESTADO INICIAL..	121
FIG. 17.	SIMULADOR APÓS 100 JOGADAS DEFINIDAS MANUALMENTE.....	121
FIG. 18.	SITUAÇÃO APÓS 279 JOGADAS E EFETIVADAS PELO AGENTE. A PROBABILIDADE DAS MÁQUINAS FOI MUDADA NA JOGADA 200.....	122
FIG. 19.	SITUAÇÃO APÓS 801 JOGADAS E EFETIVADAS PELO AGENTE.....	122
FIG. 20.	SIMULADOR PARA O PROBLEMA DE “COMER OBJETOS”.....	124
FIG. 21.	SIMULADOR PARA O CONTROLE DO “ELEVADOR”.....	127
FIG. 22.	SIMULADOR DO “JOGO DA VELHA”.....	130
FIG. 23.	SIMULADOR DO “JOGO DA VELHA” NA FASE DE TREINAMENTO SUPERVISIONADO.....	130
FIG. 24.	AGENTE JOGANDO CONTRA UM ADVERSÁRIO HUMANO.....	132
FIG. 25.	ALGUMAS DAS ESTRATÉGIAS DE JOGO APRENDIDAS PELO AGENTE.....	132

LISTA DE TABELAS

TABELA 1.	EXEMPLO DE PARAMETRIZAÇÃO DE METAESQUEMAS DE TRANSIÇÃO.....	10
TABELA 2.	ESCALA HIPOTÉTICA DE AVALIAÇÃO DE GRAUS DE INTELIGÊNCIA DE AGENTES COGNITIVOS	27
TABELA 3.	EXEMPLOS DE VOCABULÁRIO COERENTE.....	41
TABELA 4.	EXEMPLO DE PROGRAMAÇÃO DE UM CONJUNTO DE METAESQUEMAS DE TRANSIÇÃO.....	64
TABELA 5.	EXEMPLO DE TRANSIÇÕES OBSERVADAS POR UM METAESQUEMA.....	75
TABELA 6.	ESQUEMAS DE TRANSIÇÃO OBSERVADOS A PARTIR DAS TRANSIÇÕES INDIVIDUAIS MOSTRADAS NA TABELA 5	76
TABELA 7.	PSEUDOCÓDIGO PARA MANUTENÇÃO DE ESQUEMAS.....	77
TABELA 8.	TRANSIÇÕES OBSERVADAS NO PROCESSO DE TREINAMENTO.....	81
TABELA 9.	PSEUDOCÓDIGOS PARA DEFINIÇÃO DE OBJETIVOS DE SAÍDA DE UM MET.....	86
TABELA 10.	PROGRAMA DE CONTROLE DO AGENTE.....	93
TABELA 11.	TRANSIÇÕES OBSERVADAS PELO PRIMEIRO MET.....	94
TABELA 12.	TRANSIÇÕES OBSERVADAS PELO SEGUNDO MET.....	94
TABELA 13.	TRANSIÇÕES OBSERVADAS PELO TERCEIRO MET.....	94
TABELA 14.	TRANSIÇÕES OBSERVADAS PELO QUARTO MET.....	94
TABELA 15.	TRANSIÇÕES OBSERVADAS PELO QUINTO MET.....	95
TABELA 16.	TRANSIÇÕES OBSERVADAS PELO SEXTO MET.....	95
TABELA 17.	TRANSIÇÕES OBSERVADAS PELO SÉTIMO MET.....	95
TABELA 18.	PROGRAMA DE CONTROLE DO AGENTE.....	97
TABELA 19.	INDICADORES DE ROBUSTEZ DA MAH OBTIDOS EXPERIMENTALMENTE.....	108
TABELA 20.	PROGRAMA DE CONTROLE DO PROBLEMA DA MÁQUINA DE CAÇA-NÍQUEIS.....	123
TABELA 21.	TRANSIÇÕES OBSERVADAS PELO MET NAS MÁQUINAS DE CAÇA-NÍQUEIS.....	123
TABELA 22.	PROGRAMA DE CONTROLE DO PROBLEMA DA MÁQUINA DE CAÇA-NÍQUEIS.....	125
TABELA 23.	TRANSIÇÕES OBSERVADAS PELO PRIMEIRO MET.....	125
TABELA 24.	TRANSIÇÕES OBSERVADAS PELO SEGUNDO MET.....	126
TABELA 25.	PROGRAMA DE CONTROLE DO AGENTE.....	127
TABELA 26.	TRANSIÇÕES OBSERVADAS PELO PRIMEIRO MET.....	128
TABELA 27.	TRANSIÇÕES OBSERVADAS PELO SEGUNDO MET.....	128
TABELA 28.	PROGRAMA DE CONTROLE DO AGENTE.....	131
TABELA 29.	TRANSIÇÕES OBSERVADAS PELO PRIMEIRO MET.....	133
TABELA 30.	TRANSIÇÕES OBSERVADAS PELO SEGUNDO MET.....	133
TABELA 31.	POSSIBILIDADE DE ATINGIR OBJETIVOS VERIFICADA PELO SEGUNDO MET.	134

Capítulo 1

Introdução

1.1 A Busca por Sistemas “Inteligentes”

A inteligência humana foi sempre um fenômeno muito pesquisado e estudado por diversas áreas científicas. As primeiras incursões para compreender e modelar esse fenômeno datam dos tempos de Platão [1], sendo, desde então, um tema constantemente estudado na filosofia. Com o advento dos sistemas computacionais, foi possível passar da especulação à prática, sendo iniciada um grande esforço de pesquisa visando incorporar, nessas máquinas, um comportamento dito “inteligente”.

Neste contexto, na década de 50, surgem os primeiros trabalhos de Turing [2] prevendo a possibilidade de simular, artificialmente, a “inteligência”. O sistema computacional proposto de Turing é uma máquina universal capaz de emular qualquer máquina, bastando, para isso, que se escrevam os programas adequados. Segundo Turing, seria possível, portanto, desenvolver máquinas pensantes, partindo-se do pressuposto de que o cérebro humano poderia ser simulado por uma máquina discreta, ainda que extremamente complexa. Turing era um visionário e especulava sobre a possibilidade de produzir uma máquina “inteligente” que pudesse evoluir de forma similar à mente humana e que não fosse diretamente programada, mas sim submetida a um processo de “educação”.

Em 1956, matemáticos, psicólogos e engenheiros de sistemas, reunidos em uma conferência no Dartmouth College, conceberam a expressão “Inteligência Artificial” (IA) dentro de um “Paradigma de Processamento Simbólico”, no qual pensar é o mesmo que representar, calcular e manipular símbolos. Esse paradigma está baseado no que Newell e Simon [3] definiram em 1976 como hipótese do sistema de símbolos físicos:

“O sistema de símbolos físicos consiste em um conjunto de entidades, denominadas símbolos, que são padrões físicos e que podem ocorrer como componentes de outro tipo de entidade denominada expressão (ou estrutura de símbolos). Dessa forma, a estrutura de símbolos é composta de diversos casos (ou fichas) de símbolos relacionados

de alguma forma física (como uma ficha estar junto a outra). A qualquer instante, o sistema conterà uma coleção dessas estruturas de símbolos. Além dessas estruturas, o sistema também contém uma coleção de processos que operam sobre as expressões para produzir outras expressões: processos de criação, modificação, reprodução e destruição. O sistema de símbolos físicos é uma máquina que produz através do tempo uma coleção evolutiva de estrutura de símbolos. Tal sistema existe em um mundo de objetos mais amplos do que as próprias expressões simbólicas.”

Podemos notar que esta abordagem é muito mais simples do que a proposta por Turing, dessa forma, apesar do nome, a Inteligência Artificial sempre visou muito mais a resolução de problemas complexos, através de técnicas e heurísticas de programação, do que a tentativa de criação de sistemas efetivamente “inteligentes”.

Os resultados inicialmente obtidos na área de IA foram muito promissores, pois programas com heurísticas simples eram capazes de simular comportamentos e respostas que pareceriam “inteligentes” a um observador menos atento. Além disso, para problemas com um espaço de solução restrito (jogos de tabuleiro, por exemplo), técnicas de busca exaustiva aliadas a grandes velocidades de processamento permitiram a criação de sistemas de IA que tiveram desempenho superior ao dos seres humanos, como é o caso do sistema *Deep Blue* concebido pela IBM, o qual derrotou o campeão mundial de Xadrez, Gary Kasparov no ano de 1997.

Apesar do grande desenvolvimento observado nas últimas décadas da capacidade de processamento e das técnicas de programação dos sistemas computacionais, o aprimoramento das técnicas de IA foi relativamente limitado no que se refere ao desenvolvimento de sistemas “inteligentes”. As idéias de Turing tiveram alguns críticos muito severos, como Searle [4,5] que afirma que nunca poderemos desenvolver sistemas “inteligentes” apresentando o paradigma da “sala chinesa”, no qual homens que falam somente inglês processam listas de instruções a fim de manipularem textos em chinês. Para Searle [6], esses homens nunca poderão “pensar” em chinês, nem entendem o conjunto dos símbolos chineses que manipulam.

Outros autores como Law [7] e Harna [8] são mais otimistas. Para eles, o fato de sistemas “inteligentes” não terem sido ainda efetivados não representa uma impossibilidade de obtê-los, mas sim que as abordagens preponderantes dos sistemas de IA, baseadas no paradigma de processamento simbólico, levam a um beco sem saída no

caminho da obtenção deste tipo de sistema. Essa questão também é abordada por Pfeifer [9], que afirma:

“O problema fundamental com os sistemas clássicos (como os sistemas especialistas) é que eles não interagem com o mundo de uma forma direta, sempre existe um ser humano para interpretar os resultados do programa.”

Nas últimas duas décadas, a área de IA experimentou um crescimento do conexionismo como um novo paradigma. Nesse paradigma, sistemas de processamento são baseados em elementos obtidos a partir de modelos simplificados dos neurônios humanos, implementando redes neurais artificiais [10,11] (RNAs). Essas redes podem realizar tarefas de processamento bastante complexas sem a necessidade de uma predefinida programação, apenas requerendo um processo de treinamento adequado [12]. Atualmente os sistemas conexionistas têm sido cada vez mais pesquisados. No entanto, podemos constatar que a área de processamento de sinais através de RNAs ainda está muito pouco explorada. Alguns trabalhos encontrados na literatura como, por exemplo, o sistema “Darwin III”, proposto por Reeke e Edelman [13], e o sistema “DICERN”, proposto por Miikkulainen [14], indicam que RNAs de múltiplas camadas podem ser utilizadas com sucesso para compor sistemas que executam tarefas complexas como o controle de robôs e o processamento de linguagem natural sem utilizar praticamente nenhuma programação simbólica.

Neste mesmo contexto, novas técnicas que diminuem a interferência de programadores humanos vêm sendo desenvolvidas, como, por exemplo, os algoritmos genéticos [15] e a programação evolucionária [16]. Além disso, uma nova área da IA vem sendo desenvolvida visando à integração de programas computacionais com o mundo real através do conceito de “agentes autônomos” [17] e “vida artificial” [18]. Esses agentes são seres artificiais que “vivem” em mundos reais ou virtuais, recebendo informações sensoriais e atuando sobre objetos do ambiente onde estão inseridos. Nas pesquisas sobre agentes autônomos, uma grande quantidade de robôs experimentais estão sendo desenvolvidos, visando o aprimoramento dos sistemas sensoriais e dos sistemas de controle. Redes neurais artificiais vêm sendo utilizadas com sucesso para controle de robôs “insetos” [19]. No entanto, sistemas mais complexos como, por exemplo, o projeto COG [20], que visa simular, em um robô dotado de braços, visão e audição, o

desenvolvimento intelectual de um ser humano no estado inicial de vida, estão ainda utilizando técnicas de IA considerando o paradigma de processamento simbólico.

1.2 Necessidade de Sistemas “Inteligentes”

Observamos que as abordagens tradicionais dos sistemas de IA tornam-se restritivas em aplicações complexas, principalmente as que lidam com o mundo real. As abordagens normalmente utilizadas em sistemas computadorizados apresentam diversos problemas que limitam sua aplicabilidade, como:

- a necessidade de um programador humano que “empreste” parte de sua “inteligência” para definir o comportamento da máquina;
- as formas de aprendizado pela própria máquina são muito restritas e limitadas;
- os sistemas criados dentro desta abordagem são simples manipuladores de símbolos, que nunca terão nenhum sentido concreto para a máquina.

Por outro lado, se considerarmos a proposta de Turing para um sistema artificial “inteligente” que aprenda de forma análoga a um ser humano, podemos acreditar que esse sistema irá apresentar características similares aos processos intelectuais humanos. Essas “Inteligências Artificiais” terão habilidades cognitivas e capacidade de aprendizado praticamente indistinguíveis das dos seres humanos, podendo ser criativas e até mesmo apresentar alguns tipos de emoções. Essas máquinas teriam a capacidade de lidar com novos problemas e seriam extremamente úteis na medida em que fossem capazes de lidar com o mundo real, podendo ser empregadas em diversas funções, tais como:

- controle de robôs industriais;
- controle de veículos e aeronaves;
- gerenciamento de sistemas de informações e sistemas administrativos;
- tradução simultânea de texto e fala;
- apoio a especialistas nas mais diversas áreas.

Além disso, se estas “Inteligências Artificiais” tivessem domínio de linguagem natural, tanto escrita quanto falada, poderiam ser empregadas em inúmeras tarefas de atendimento ao público e também como assistentes pessoais.

Consideremos, por exemplo, a área de ensino: o tipo de “Inteligência Artificial” que vislumbramos poderia atuar como um “professor virtual”, conversando com seus alunos em linguagem natural, podendo esclarecer todas as dúvidas e adaptando-se ao ritmo de cada aluno de forma semelhante e talvez indistinguível à de um professor humano. Esse tipo de professor virtual seria praticamente imortal e poderia trabalhar ininterruptamente com dezenas (ou talvez milhares) de alunos. Através desse exemplo, não é difícil imaginar as vantagens desse tipo de sistema “inteligente”, bem como os impactos positivos (ou negativos) que essa tecnologia poderia trazer se for bem (ou mal) aplicada.

Ao considerarmos sistemas “efetivamente inteligentes” estamos preocupados fundamentalmente com capacidades cognitivas de comunicação, criatividade e aprendizagem, comparáveis às dos seres humanos. Em nossa visão, nesse contexto, não são tão importantes outras características normalmente associadas à “inteligência” humana, tais como: autoconsciência, emotividade e espiritualidade, que também possam vir a existir em sistemas “inteligentes”. Assim, a questão de um “professor virtual”, ou seja, cujas características “externas” sejam indistinguíveis das de um “professor real”, ser efetivamente “inteligente” ou apenas simular essa “inteligência” pode ser normalmente desconsiderada. Esse tipo de questão torna-se relevante apenas em situações limites, como no caso desse mesmo “professor virtual” entrar em “depressão” ao ficar consciente de que pode “morrer” se for desligado.

Por fim, os aspectos éticos, morais e de segurança associados a um sistema artificial efetivamente “inteligente” são muito complexos, como nos mostram as inúmeras obras de ficção científica que tratam desse tema. Considerando apenas como exemplo os danos que um especialista humano (*hacker*) pode causar ao quebrar as barreiras de segurança de sistemas conectados à Internet, podemos imaginar o caos que uma “IA *hacker*” poderia gerar. Assim, antes de conectarmos sistemas “inteligentes” na Internet ou de colocá-los no comando de máquinas ou processos críticos, teremos de dotá-los de mecanismos de segurança que poderão ser semelhantes às “três leis da robótica” propostas por Isack Asimov [21], ou talvez mecanismos tão distantes de nossa realidade atual, que nem os melhores escritores de ficção científica os tenham imaginado ainda.

1.3 A Plausibilidade dos Sistemas “Inteligentes” Artificiais

Em nosso entendimento, os argumentos contra a possibilidade de desenvolvimento de sistemas de processamento de informações efetivamente “inteligentes” não se sustentam pelo simples fato de já existir um desses sistemas, que é o próprio cérebro humano. Para demonstrar a possibilidade da existência de um sistema “inteligente” implementado em um computador, podemos imaginar um cenário onde fôssemos capazes de construir um simulador extremamente poderoso, em que pudéssemos modelar exatamente o comportamento de uma célula neural humana, simulando todas as reações físicas e químicas que ocorrem em um neurônio. Se imaginarmos um simulador bilhões de vezes mais poderoso, conseguiremos simular todos os neurônios de um cérebro humano bem como os sistemas sensoriais e motores a ele associados. Essa simulação de cérebro poderia ser educada da mesma forma com que educamos um bebê humano, vindo a se tornar um ser artificial “inteligente”, sem que os bilhões de processadores individuais necessários para implementar o simulador pudessem ser chamados de inteligentes.

Segundo Kaku [22], no ritmo de evolução de nossos sistemas computacionais, que dobra a cada 18 meses, poderá ser viável, em algumas décadas, que uma única máquina tenha capacidade de processamento milhares de vezes maior do que a dos computadores atuais, aproximando-se, e talvez até ultrapassando, a capacidade de processamento de um cérebro humano.

Alguém poderia criticar esta hipótese, afirmando que, mesmo de posse de um simulador adequado, nunca poderemos mapear totalmente as conexões de um cérebro humano para alimentar esse simulador. Rebatemos essa crítica afirmando que todos os cérebros humanos são construídos com base na informação contida nos filamentos de DNA de uma única célula fecundada e que esse DNA, que já foi completamente mapeado através do projeto GENOMA, um dia também poderá ser inteiramente compreendido.

Assim, se pudermos simular todos os processos físico/químicos de uma única célula fecundada, em nível molecular (ou mesmo atômico), não estaremos muito longe de simular seu processo de crescimento e duplicação, que culmina na formação de cada uma das seis trilhões de células [22] que, reunidas, compõem o corpo humano.

Ainda que o caminho para um “simulador de cérebro” represente uma possibilidade real de criação de um sistema artificial efetivamente “inteligente”, não

acreditamos que esse caminho venha a ser efetivamente trilhado, pois demandaria capacidades enormes de processamento sem nos dar a garantia de efetivamente entender como um cérebro funciona.

Estimamos que os sistemas “inteligentes” devam surgir da criação de novas técnicas de programação, de novos modelos de sistemas e principalmente de novos paradigmas de processamento de sinais.

1.4 Necessidade de Novos Paradigmas de IA

Ao iniciar o presente trabalho, tínhamos a noção de que o caminho para a obtenção de sistemas efetivamente “inteligentes” passa pela quebra de uma série de paradigmas, principalmente, do paradigma de processamento simbólico. Isso representa uma mudança de enfoque no qual “programas manipuladores de símbolos” serão substituídos por “agentes autônomos” ou “agentes cognitivos” que efetivamente “vivam” em mundos reais ou artificiais, desenvolvendo alguma forma de “compreensão” desses mundos.

Consideremos um simples exemplo dado pela sentença “O bolo é doce”. No modelo de processamento simbólico podemos construir programas que manipulem esse tipo de sentença, mas essa sentença nunca terá um significado concreto para a máquina. Já, no contexto de um “agente cognitivo”, a “compreensão” dessa sentença demanda no mínimo que o agente viva em um mundo onde exista o objeto “bolo”, que possua uma série de atributos dentre os quais esteja o grau de “doçura”, que possa ser captado por sensores do agente quando esse interagir com o “bolo” (“comer”). Mesmo com todos esses elementos, não sabemos se um “agente cognitivo” poderá realmente “compreender” a sentença “o bolo é doce” como um ser humano a compreende, mas estamos certos de que sem tais elementos nenhuma “compreensão” poderá ser atingida.

Uma argumentação contrária a esta posição é a de que sistemas de controle de “agentes autônomos” não passam de um programa seqüencial baseado em uma série de regras de manipulação de dados e, dessa forma, tudo que eles realizam também poderia ser realizado através de técnicas tradicionais de programação. Esse tipo de argumentação reducionista tem gerado muitos questionamentos errôneos do tipo: “por que pesquisar redes neurais se tudo o que elas fazem pode ser realizado com uma máquina seqüencial?”

ou “por que estudar redes neurais com múltiplas camadas se uma rede de três camadas pode representar o processamento de qualquer rede maior?”. O fato é que, nesses casos, vale a máxima “o todo é maior do que a soma das partes”, pois esses argumentos não levam em conta diversos aspectos práticos sobre a forma como os sistemas são construídos, programados ou treinados.

Citamos, como exemplo, o sistema “DICERN”, proposto por Miikkulainen, que é composto por aproximadamente 10 mil neurônios agrupados em mais de 20 camadas e que realiza a proeza de armazenar frases e responder a questões sobre essas frases, formuladas em linguagem natural. Isso sem que nenhum processamento simbólico seja necessário. Talvez o sistema “DICERN” pudesse ser modelado por uma rede neural de 3 camadas, mas o número elevado de neurônios e a complexidade do processo de treinamento de tal rede iriam certamente inviabilizar a sua aplicação prática.

Consideremos também um problema complexo como a tradução de frases entre dois idiomas. Em uma visão reducionista, podemos resolvê-lo facilmente montando uma tabela com o pareamento de todas as frases possíveis nos dois idiomas, porém essa tabela teria de possuir aproximadamente 10^{60} frases e dificilmente poderia ser efetivamente construída. Em uma abordagem de IA tradicional, construiríamos uma tabela associando apenas palavras em um idioma com conjuntos de palavras do outro, utilizando uma série de heurísticas para escolher o melhor significado. No entanto, quem já utilizou esse tipo de programa sabe o quanto eles são limitados. Assim, uma forma de se obter uma tradução equivalente à realizada por um perito humano seria através de um sistema artificial que efetivamente tivesse o domínio sintático e semântico dos dois idiomas.

1.5 Um Novo Paradigma de Controle e Processamento de Sinais

No desenvolvimento da presente tese, constatamos que os paradigmas de processamento simbólicos e conexionistas que aparentam ser bastante distintos, na realidade estão inseridos em um contexto que é muito mais elementar. Os sistemas simbólicos são modelados por uma máquina seqüencial que lê padrões de dados (ou instruções), processa esses dados e gera novos dados na saída. São, em essência, similares aos sistemas conexionistas, pois uma rede neural também recebe padrões (dados) de entrada, processa os mesmos e gera padrões de saída. Isso ocorre, pois os dois sistemas

estão baseados no paradigma tradicional de controle e processamento de sinais, que opera segundo o modelo “entrada - processamento - saída”. Esse paradigma está de tal forma inserido em todas as áreas da ciência, como: engenharia, computação, economia, psicologia, e outras, que muitas vezes pode passar despercebido. De fato, a seqüência “entrada - processamento - saída” nos parece tão natural que nem percebemos que ela não é a única possível. Isso se deve ao fato de que os conceitos de “entrada” e “saída” são tão elementares que nos parecem conceitos que se sustentam por si só, sem fazer parte de um paradigma.

A nova visão de controle e processamento de sinais, proposta nesta tese, foi denominada processamento de “objetivos”. Nessa abordagem, não se aplicam conceitos de “entradas” e “saídas” como conhecemos tradicionalmente (a não ser nas interfaces mais externas dos sistemas). Nesse modelo, os sinais manipulados são tratados segundo relações de causa e efeito, sem levar em conta o fato de cada sinal individual ser efetivamente uma “entrada” ou “saída”. Além disso, surge uma nova classe de “sinais virtuais”, denominados “sinais objetivos” (estes sim podem ser “objetivos de entrada” ou “objetivos de saída”), que obedecem a novas regras de processamento e de atribuição de valores. Por exemplo, múltiplos “objetivos de saída” podem estar conectados a um único “objetivo de entrada” e um único “objetivo” pode assumir diversos valores simultâneos ou ainda valores abstratos como “nenhum valor” ou “objetivo atingido”. Por fim, o “processamento de objetivos” lida com múltiplos espaços de “causa” e “efeito” (que podem ser definidos de forma não causal) modelados por metaesquemas de transição hierarquizados de forma a propagar os “objetivos” através desses espaços.

A veracidade e aplicabilidade da abordagem de “processamento de objetivos” será demonstrada e discutida na presente tese, sendo que os primeiros resultados obtidos já podem ser considerados muito animadores. Como exemplo, citamos o programa do “Jogo da Velha” que é um modelo hierárquico de interligação de metaesquemas de transição, contendo as informações mostradas na Tabela 1.

TABELA 1
EXEMPLO DE PARAMETRIZAÇÃO DE METAESQUEMAS DE TRANSIÇÃO

(\$CC, E1, E2, E3, E4, E5, E6, E7, E8, E9 E11 E12 O1, O2, O3, O4, O5, O6, O7, O8, O9 0 1 1 0)
(S1, E10, E14 E1 S1 O1)
(S2, E10, E14 E2 S2 O2)
(S3, E10, E14 E3 S3 O3)
(S4, E10, E14 E4 S4 O4)
(S5, E10, E14 E5 S5 O5)
(S6, E10, E14 E6 S6 O6)
(S7, E10, E14 E7 S7 O7)
(S8, E10, E14 E8 S8 O8)
(S9, E10, E14 E9 S9 O9)

Observando a seqüência de letras da Tabela 1, cujo significado será apresentado no decorrer desta tese, podemos constatar alguns efeitos advindos da definição deste novo paradigma:

- definição de uma nova forma de programação com filosofia diferente de todas já conhecidas;
- possibilidade de resolver problemas de uma certa complexidade como é o “Jogo da Velha”, com uma quantidade mínima de informação inicial.

Acreditamos que um longo caminho ainda deverá ser percorrido na busca de sistemas de IA efetivamente “inteligentes”, e que esta tese representa mais um pequeno passo na direção desse objetivo.

1.6 Histórico do trabalho

A meta inicial do presente trabalho foi a criação de novos modelos que auxiliassem na obtenção de sistemas artificiais efetivamente “inteligentes”. Para tentar atingir essa meta, buscamos subsídios não só nos modelos da área de IA tradicional, mas também em modelos desenvolvidos nas áreas de ciências humanas, principalmente no modelo semiótico proposto por Charles Pierce [23] e no modelo construtivista proposto por Jean Piaget [24]. Nesse contexto, nosso principal objetivo passou a ser o desenvolvimento de um “agente cognitivo” que, “vivendo” em um mundo virtual, fosse capaz de aprender uma série de habilidades e comportamentos “inteligentes”.

Logo no início do trabalho nos deparamos com um problema básico para o qual não encontramos uma solução específica em literatura: a falta de mecanismos de avaliação para verificar o desempenho de um dado sistema quanto ao seu nível de “inteligência”. Em uma analogia a esse problema, podemos considerar o caso em que um pesquisador tentasse estudar terremotos sem dispor de qualquer escala que permitisse comparar suas magnitudes, mesmo de forma relativa. Assim surgiu a questão: como estudar sistemas “inteligentes” sem uma escala que permita quantificar a inteligência de um sistema?

Os testes encontrados na literatura são muito complexos e normalmente definem apenas dois níveis (inteligente e não inteligente), além do mais, os sistemas propostos testados são sempre programados para resolver problemas específicos, como controlar um veículo ou executar uma determinada tarefa complexa. No nosso ponto de vista, um sistema efetivamente “inteligente” deve aprender a lidar com uma série de problemas sem ser programado nos mínimos detalhes para cada um deles. Tais sistemas seriam passíveis, portanto, de serem avaliados por meio de uma série padrão de problemas de nível de complexidade crescente, compondo assim uma escala para medição de “inteligência”. O fato desse tipo de escala não estar disponível só demonstra o quão incipiente ainda está a área de IA que trata de sistemas efetivamente “inteligentes”. Assim, decidimos que nossa primeira contribuição seria propor a concepção de dois tipos de escalas, mostrando exemplos de como elas poderiam ser implementadas.

Logicamente para ter valor prático, tais escalas deveriam ser amplamente difundidas e testadas, para serem então validadas. Tal objetivo demandaria um esforço conjunto de muitas entidades de pesquisa colaborando nessa tarefa, o que, evidentemente, está além do escopo desta tese de doutorado. Assim, contentamo-nos em divulgar a necessidade e um caminho para a realização desse tipo de ferramenta de avaliação, sabendo que, por ser tão necessário à evolução da área de IA, ele um dia acabará sendo estabelecido.

Após a definição destas escalas, procuramos atacar mais diretamente os objetivos propostos, considerando duas vertentes básicas. A primeira consiste na interação de um agente cognitivo com um ambiente virtual, criando o trinômio “símbolo - intérprete - significado” proposto por Pierce [23] no contexto da semiótica. A segunda se apóia nos modelos de aprendizado construtivista definidos por Piaget [24]

dentro de um contexto humano, principalmente, o modelo de representação de informações através de “esquemas mentais”.

O modelo de Piaget é muito mais filosófico do que matemático e se baseia também em um trinômio dado por “situação de aplicação do esquema - ação a ser realizada - resultado esperado” que pode também ser representado pelo modelo de controle tradicional “entrada - processamento - saída”.

Partindo do modelo de esquemas mentais de Piaget, refletimos muito sobre como as informações são tratadas em nosso cérebro e percebemos que trabalhamos muito mais com modelos mentais de “causa e efeito” do que com modelos de “entrada e saída”. Assim, definimos um novo modelo de processamento baseado em “metaesquemas de transição”, que é uma generalização do modelo tradicional, sendo portanto muito mais abrangente.

Os metaesquemas de transição não operam somente sobre sinais de entrada e saída, mas também manipulam uma nova classe de sinais virtuais que denominamos “sinais objetivos”.

Os metaesquemas de transição são essencialmente hierárquicos e podem ser facilmente interligados, permitindo que um problema complexo seja naturalmente subdividido em uma série de problemas simples.

A aplicação prática deste novo modelo demandou a criação de uma máquina de inferência para o processamento de metaesquemas. Essa máquina foi aplicada ao controle de um agente cognitivo interagindo com um mundo virtual através de uma série de sensores e atuadores.

Além disso, criamos uma “sintaxe” para configuração de metaesquemas de transição, que permite definir de forma textual uma estrutura composta de uma série de metaesquemas. É bom lembrar que essas estruturas de metaesquemas sempre estão associadas ao controle de um agente cognitivo que aprende de forma supervisionada (treinamento com reforço) a resolver um dado problema definido no mundo no qual ele está inserido. Para testar esse modelo, foram definidos vários problemas com um grau de complexidade crescente, cada um representando uma situação distinta e sendo tratado por um dado tipo de agente cognitivo e por uma dada hierarquia de metaesquemas.

Estas aplicações apresentaram resultados muito interessantes como, por exemplo, o agente que interage com um “Jogo da Velha”. Ele demonstrou características e

aprendizado muito similares às de um jogador humano. Em determinada altura do processo de aprendizagem, o agente passa a utilizar as habilidades observadas em seu oponente em benefício próprio, tornando-se, no mínimo, tão hábil quanto este. Outra característica interessante é que não sabemos *a priori* o que o agente irá fazer em uma dada situação, a não ser executando a máquina de inferência de metaesquemas que o controla. Por fim, podemos observar que os agentes puderam ser treinados tanto de forma supervisionada como não supervisionada, podendo o treinador assumir momentaneamente o controle dos atuadores do agente para ensinar mais rapidamente uma dada tarefa, de maneira análoga a de um professor que segura na mão do aluno quando o está ensinando a escrever.

Outra característica do novo modelo é que, no treinamento com metaesquemas, ao contrário dos treinamentos com reforço tradicional nos quais os reforços positivos e negativos normalmente alimentam uma estrutura de controle de aprendizado dissociada do agente, o agente normalmente passa a ter dois objetivos máximos: “sentir prazer” e “não sentir dor”. Levando-se para a área de psicologia, levantamos então a questão: não são estes dois os objetivos maiores que movem os seres humanos em suas existências?

Da tentativa de utilização do modelo proposto para tarefas relacionadas ao processamento de informação simbólica, tal como definido na semiótica de Pierce [23], percebemos algumas limitações e a necessidade de utilização de algum tipo de memória interna ao agente associada ao mecanismo de controle composto por metaesquemas.

Logo, percebemos que deveríamos utilizar neste processo algum tipo de memória associativa, mas infelizmente todas as memórias associativas que encontramos em literatura apresentavam grandes problemas, principalmente quanto a processos de treinamentos que inviabilizavam sua utilização no contexto do agente autônomo por nós idealizado.

Assim, buscamos desenvolver um novo tipo de memória associativa, a qual denominamos memória associativa holográfica. Essa nova estrutura de memória é capaz de armazenar pares ou seqüências de informações simbólicas, sem necessidade de treinamento, recuperando-as em um único passo através mesmo de um fragmento da informação armazenada.

A memória holográfica proposta trata informações ambíguas e pode também “esquecer” informações mais remotas ou informações não corretas, sendo também bastante robusta à perda de elementos e à adição de níveis elevados de ruído.

A eficiência da memória holográfica é relativamente baixa, e, em um caso típico, uma memória associativa composta de uma matriz de 100×100 elementos pode armazenar pouco mais do que 70 pares de informação. Essa deficiência, que é praticamente uma característica universal das memórias associativas, é o preço que pagamos pela sua robustez.

É interessante observar que tal estrutura de memória pode ser diretamente implementada através de redes neurais artificiais e que suas características de operação nos parecem muito similares às das inferidas para as memórias humanas, servindo para nos auxiliar a compreender um pouco o funcionamento de nossa mente.

Aplicamos a memória associativa holográfica proposta em um sistema de manipulação de informações simbólicas e observamos que ela foi capaz de responder com sucesso à questão sobre os dados armazenados, sem recorrer a qualquer processo tradicional de busca de informações.

Esperávamos como último passo deste trabalho aplicar conjuntamente a memória holográfica ao agente controlado por metaesquemas no tratamento de informações simbólicas, obtendo um agente que compreendesse frases do tipo “o bolo é doce”. Essa tarefa todavia se mostrou mais complexa do que pensávamos em princípio, extrapolando o escopo da presente tese.

1.7 Organização do Trabalho

No Capítulo 2, são apresentados alguns conceitos básicos usados neste trabalho, sendo abordados, de forma resumida, os conceitos de Semiótica e Construtivismos propostos por Pierce [23] e Piaget [24], respectivamente, bem como a definição de agentes cognitivos. Consideramos que os demais conceitos, associados à área de IA, que são algumas vezes mencionados neste trabalho, tais como sistemas especialistas e redes neurais artificiais, são básicos nesta área, com ampla literatura disponível e, dessa forma, já são de conhecimento do leitor ou poderão ser por ele facilmente acessados.

No Capítulo 3, apresentaremos o conceito de escalas para medição do grau de “inteligência” de agentes cognitivos, que consideramos de extrema importância para o aprimoramento da área de IA. No nosso entendimento, será muito difícil obter sistemas

efetivamente “inteligentes” se não tivermos formas mínimas para verificar e/ou medir certos graus de “inteligência”. Duas escalas são apresentadas a título de exemplo.

No Capítulo 4, apresentamos a principal contribuição desta tese, definindo novos modelos de processamento baseados em “Esquemas de Transição”, que permitem a aplicação de um novo tipo de treinamento com reforço, no qual tarefas maiores são subdivididas e organizadas em tarefas menores até o limite de ações individuais. Um exemplo de controle de um agente relativamente complexo é apresentado.

No Capítulo 5, apresentamos um novo modelo de memória associativa, que apresenta características de funcionamento semelhantes às das memórias associativas biológicas. A “Memória Associativa Holográfica”, definida neste trabalho, apresenta a vantagem de poder ser treinada em um único passo, sendo muito robusta e pouco sensível à perda de elementos de armazenamento individuais e suportando altos níveis de degradação por ruído.

As discussões e conclusões gerais são apresentadas no Capítulo 6, seguidas por dois Apêndices, o primeiro, apresentando exemplos de aplicação de metaesquemas de transição e o segundo, uma análise da capacidade de armazenamento das memórias associativas holográficas.

Abordagens Semiótica e Construtivista

2.1 Introdução

Paralelamente à evolução da área de IA, nas ciências humanas, também se buscava um modelo para a inteligência e o comportamento inteligente. Dentre outros tópicos, vale a pena destacar o desenvolvimento da semiótica por Pierce e Morris [25] e o estudo do desenvolvimento da inteligência nos seres humanos por Piaget [26].

Piaget realizou um amplo estudo do processo de aprendizagem de crianças. Ele propôs uma classificação para o desenvolvimento de seres humanos baseada em uma série de etapas caracterizadas, principalmente, pela capacidade de cada ser interagir com o meio ambiente, como também a de formar agrupamentos e processos de classificação.

As teorias de Pierce visam o entendimento de como os seres humanos manipulam símbolos (ou signos). Para Pierce, os signos não existem isolados na mente, mas sim na forma de uma tríade, devendo ser considerados os objetos (associados a um contexto de mundo), os próprios símbolos (associados a uma forma de representação dos mesmos) e a interpretação mental dos símbolos (significado).

Apesar da motivação comum, a inteligência artificial, a semiótica e o construtivismo, como disciplinas científicas, seguiram desenvolvimentos distintos. A inteligência artificial partiu para a criação de estruturas matemáticas que emulassem características particulares da inteligência. Os modelos utilizados na semiótica e no construtivismo, sendo modelos descritivos, descrevem de modo mais subjetivo o comportamento inteligente, utilizando conceitos e nomenclaturas que não encontram um suporte imediato na matemática. Dada essa diferenciação, observa-se que os modelos originados na inteligência artificial têm uma formulação matemática mais completa, ao passo que os modelos provenientes da semiótica e do construtivismo são mais generalistas, apresentando conceitos que muitas vezes não têm uma formulação matemática diretamente associada. Por esse motivo, os modelos oriundos da semiótica e do construtivismo

conseguiram avançar mais profundamente na explicação dos fenômenos que compõem a inteligência, ou seja, são capazes de explicar mais facetas da inteligência do que os modelos tradicionais de inteligência artificial. Dessa forma, um ponto importante deste trabalho é a busca de modelos teóricos não somente nas áreas de ciências exatas, mas também nas áreas de ciências humanas, principalmente nos modelos derivados da semiótica e do construtivismo.

2.2 Abordagem Semiótica

A semiótica, proposta por Pierce [27], define estruturas de conhecimento compostas por três elementos: signo, objeto e interpretante. Essa tríade pode também ser mapeada nos modelos da IA, conforme ilustrados pela Fig. 1.

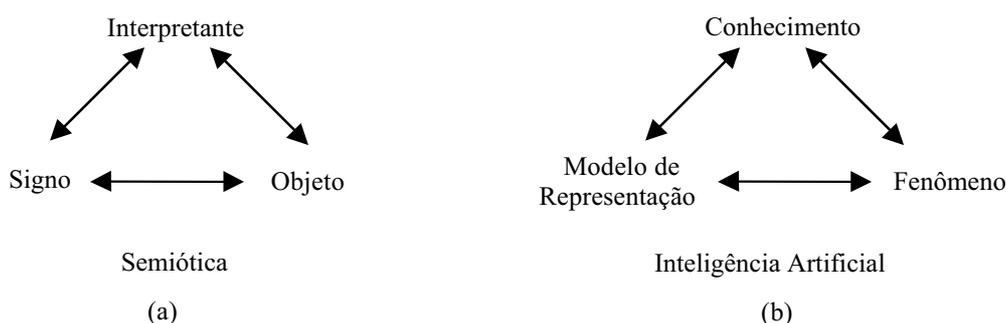


Fig. 1. Mapeamentos: (a) Semiótica e (b) Inteligência Artificial.

Na tríade da semiótica, o signo é utilizado para representar o objeto, cuja compreensão por uma mente inteligente corresponde ao interpretante. Ou seja, o interpretante é a intelectualização do objeto. Na tríade da IA, um fenômeno do ambiente (objeto), interpretado como um conhecimento a respeito do ambiente, é representado por um modelo de representação do conhecimento, que corresponde ao signo.

No modelo proposto por Pierce, todos estes elementos devem estar presentes e, dentro desta ótica, é impossível considerar uma interpretação simbólica fora de um contexto de ambiente. É interessante observar o fato de que uma proposição lançada no início do século XX no contexto da psicologia e da lingüística seja diretamente aplicável a um contexto de inteligência artificial e, mais do que isso, aborde diretamente um ponto

que, para Law [6] e Pfeifer [8], é de fundamental importância para o desenvolvimento de sistemas “inteligentes”: a interação direta com o ambiente.

Na abordagem da semiótica, alguns elementos são considerados essenciais, a saber :

- objetos;
- signos (símbolos);
- interpretantes.

Os objetos têm uma série de propriedades físicas que os definem, além disso, estão inseridos em um contexto. Algumas propriedades dos objetos podem ser detectadas por meio de sensores. Dessa forma, fica claro que os objetos devem estar inseridos em um ambiente dentro de uma certa disposição espacial e temporal.

2.3 Abordagem Construtivista

A ciência, definida e estudada por Piaget [28,29], foi a Epistemologia Genética, que trata da gênese (mecanismos formadores) da natureza e da evolução dos processos cognitivos.

Vinculando a psicologia à biologia, Piaget viu na inteligência a etapa final do processo de adaptação dos seres vivos ao meio ambiente, definindo, assim, um modelo no qual a inteligência seria o resultado de uma construção de esquemas mentais que se daria ao longo do período de desenvolvimento de uma criança, do nascimento até o fim da adolescência. Esse modelo passou então a ser denominado Modelo Construtivista de Piaget.

Até Piaget, apesar da função básica da mente ser o “pensar”, a única coisa que a psicologia não pesquisara era o pensamento. Para os behavioristas, o pensamento era uma “caixa preta”, inacessível à pesquisa. Piaget quebrou esse paradigma e dedicou-se, precisamente, a descobrir como o ser humano pensa, por que uns são lógicos e outros não, por que uma criança pensa diferente de um adulto, etc.

O Construtivismo de Piaget entende que noções de objetos e sensações a eles associadas não são herdadas, mas abstraídas das relações da criança com o mundo exterior. De forma semelhante, as noções de lógica, classificação, números e, por fim, do que somos seriam gradualmente construídas na mente de uma criança.

Desde Aristóteles, por exemplo, todos supõem que a vida mental procede da percepção. Piaget demonstrou que essa afirmativa é falsa. A vida mental é uma dublagem da ação. Primeiro o estímulo, depois a resposta, dizem os psicólogos. Para Piaget, primeiro vem a resposta. É a resposta que busca por um estímulo. A neurobiologia vem suportar os experimentos bem sucedidos de Piaget. O estímulo só estimula se o organismo (mente) estiver preparado (necessidade) para recebê-lo.

O trabalho de Piaget, fundamento das pesquisas experimentais modernas sobre a gênese da inteligência nos indivíduos, está centrado no conceito de esquema definido por ele:

Um esquema é uma unidade de comportamento e conhecimento que interage e evolui em conjunto com o seu ambiente e outros esquemas.

Os esquemas iniciais correspondem aos reflexos. Nas crianças, confundem-se com as próprias ações. Sofisticações adicionais, envolvendo combinação de esquemas, abstrações sobre ações específicas, permitem a transcendência à ação física e o início do pensamento abstrato.

Esquemas de olhar, pegar o que se vê, esconder um objeto sob outro, empurrar um objeto com outro, são exemplos de esquemas pós-reflexo.

A inteligência seria o resultado do desenvolvimento, crescimento e diferenciação desses esquemas inatos, em um processo de elaboração que ultrapassaria os limites do nível biológico. A adaptação inteligente do indivíduo ao meio vai além da adaptação orgânica ao ambiente, atingindo o nível cognitivo, isto é, o nível das ações que implicam o conhecimento. No limite, a inteligência seria a adaptação das ações interiorizadas que constituem o pensamento. Há adaptação quando um organismo se transforma em função do meio e quando essa variação tem por efeito um acréscimo das trocas em ambos, acréscimo esse favorável à sobrevivência do organismo. A adaptação é um equilíbrio entre dois mecanismos distintos: a assimilação e a acomodação.

Assimilação corresponde à utilização, por um esquema, das coisas existentes no mundo como parte de seu próprio funcionamento; acomodação consiste na modificação de esquemas para que se ajustem às coisas novas que aparecem no mundo.

Ao longo de sua vida, Piaget observou que existem formas diferentes de interagir com o ambiente nas diversas faixas etárias. A essas maneiras típicas de agir e pensar denominou períodos, estágios e fases. Segundo ainda Piaget, observa-se dois períodos

principais:

- período 1: vai de 0 a cerca de 2 anos e é definido como estágio sensório motor;
- período 2: vai de 2 anos até os 11-12 anos e é definido como período das operações concretas.

A cada um destes períodos estão associados uma série de estágios e fases que definem de forma bastante completa a evolução do desenvolvimento cognitivo de uma criança.

Podemos observar que a Teoria Construtivista de Piaget é bastante ampla e abrangente, sendo que um dos problemas para sua completa utilização na área de IA é a definição de modelos matemáticos que possibilitem a aplicação das mesmas em um sistema computacional. Uma parte da teoria de Piaget, que possibilita esta modelagem, é a da representação de informações por meio de *esquemas de assimilação*. Os esquemas podem ser modelados por conjuntos de informações assim definidos:

- objetos e situações aos quais o esquema se aplica;
- ações a serem tomadas;
- objetivos ou resultados previstos;
- parâmetros de avaliação.

Este modelo será abordado com maiores detalhes no Capítulo 5 deste trabalho.

2.4 Agentes Cognitivos

Um “agente autônomo” [30] é uma entidade autônoma capaz de interagir com um ambiente (um mundo real ou artificial). O ambiente é povoado com objetos que apresentem um conjunto de propriedades e são manipulados pelos agentes. O conceito de agente autônomo é bastante amplo sendo aplicável desde sistemas de controle de “insetos” e de robôs até os “assistentes” que já são encontrados em diversos programas comerciais.

No contexto deste trabalho, preferimos utilizar o conceito de “agente cognitivo”, definido por Fialho [31], que mais se aproxima do modelo proposto por Pierce [27] no que pressupõe a existência de um intérprete, que pode ser modelado por um agente cognitivo. Esse agente “vive” em um ambiente recebendo informações através de uma série de

sensores e agindo sobre o meio através de um conjunto de atuadores. Dessa forma, para o agente cognitivo processar uma informação do tipo “o bolo é doce”, segundo a semiótica de Pierce, devemos ter no mínimo um intérprete (agente) que recebe informações de objetos (bolo é sabor doce) através de informações sensoriais (forma e gosto do bolo) e associa os signos (as palavras “bolo” e “doce”) aos objetos e também os interrelaciona. Esse conjunto de signos “bolo é doce” está ligado aos estímulos sensoriais que indicam que o objeto “bolo” tem a propriedade de ser “doce”, que surge da ação do agente “comer” o bolo.

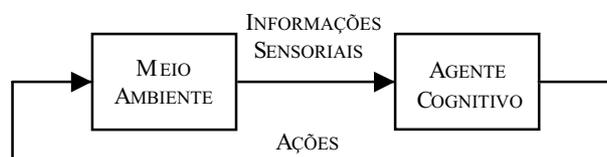


Fig. 2. Interação entre o agente cognitivo e o meio ambiente.

Nos sistemas tradicionais de AI, a agregação de conhecimento passa pela “programação de regras” realizadas por um ser humano. No caso dos agentes cognitivos, esse tipo de “programação” é substituída por um processo de aprendizagem contínuo.

Neste modelo de “agentes cognitivos”, o conhecimento inicial do agente sobre o seu mundo é muito pequeno. Podemos inferir que devam existir algumas fases distintas no processo de aprendizagem:

- inicialmente o agente aprende que existem objetos em seu mundo e que interagindo com eles pode identificar suas propriedades de forma totalmente sensorial e não simbólica;
- a seguir, o agente irá aprender a associar os símbolos com os objetos e com suas características;
- por fim, o agente deverá converter representações internas em conjuntos de símbolos que, desta forma, terão significados.

Os processos de aprendizagem dos agentes cognitivos deverão ser muito parecidos com os processos de aprendizagem humanos, englobando principalmente processos de “tentativa e erro” e processos de treinamento com reforço (punição e recompensa), podendo ser estendidos para processos de aprendizagem por imitação e repetição e, também, para treinamento através de transmissão de informações simbólicas.

2.5 Abordagem Proposta

O presente trabalho busca uma nova abordagem que consiste no desenvolvimento de agentes cognitivos baseados nos modelos propostos pela Semiótica de Pierce e pelo Construtivismo de Piaget, dentro dos quais devem ser observados alguns aspectos elementares:

- o agente deve estar inserido em um meio ambiente (virtual ou real) e deve ser capaz de receber uma série de informações sensoriais do meio e também de atuar sobre o meio;
- o agente, em princípio, não está pré-programado com informações do meio ambiente e desconhece os problemas que terá de superar para “viver” nele;
- em princípio, qualquer passagem direta de “informações” do programador para o agente não é permitida;
- o agente deve aprender continuamente, interagindo com o meio e recebendo estímulos de punição ou recompensa;
- o agente poderá aprender a manipular informações simbólicas através da interação com outros agentes;
- qualquer informação simbólica que o agente venha a manipular deverá ser associada a objetos ou ações que estejam inseridas no ambiente.

Estes pontos impõe grandes limitações no desenvolvimento dos agentes cognitivos, como mostra o exemplo do problema “jogar Xadrez”.

Para jogar Xadrez, segundo nosso modelo, o agente deveria aprender inicialmente a reconhecer a “forma” (seja ela uma imagem ou uma representação numérica) das peças e o tabuleiro, a seguir deveria aprender como movimentar cada peça e os objetivos básicos do jogo. Inicialmente o agente será um jogador “fraco” e espera-se que, com o passar do tempo, desenvolva seu jogo, elabore estratégias e torne-se, no mínimo, um “bom” jogador.

É fácil perceber que esta abordagem é muito mais complexa do que as normalmente empregadas para fazer uma máquina “jogar Xadrez”, pois, nas abordagens tradicionais, temos sempre um programador humano (ou uma equipe de programadores) que sabe jogar Xadrez e define no mínimo as regras de movimentação das peças, os objetivos do jogo e estratégias de buscas de soluções, ou seja, o programador humano “transfere” sua compreensão do problema para a máquina. Na abordagem tradicional, para

o caso do “Jogo de Xadrez”, um programador com apenas um conhecimento elementar do jogo poderá desenvolver um programa que jogue Xadrez muito bem a ponto de derrotar um experiente jogador humano. Isso é possível, pois o espaço de solução para o problema do “Jogo de Xadrez”, apesar de muito grande, pode ser exaustivamente pesquisado por uma máquina de grande capacidade de processamento.

A abordagem que propomos para o “Jogo de Xadrez” é muito mais complexa do que as tradicionais, porém, dentro dela, o agente poderia aprender qualquer jogo ou resolver qualquer problema de complexidade equivalente ao “Jogo de Xadrez” sem necessidade de uma nova “programação”, mas sim de um novo processo de treinamento.

O fato dos programas que jogam Xadrez não estarem sendo programados desta forma se deve tanto à dificuldade intrínseca desse tipo de aprendizado como também à facilidade de resolver o problema do “Jogo de Xadrez” pelos métodos heurísticos tradicionais. Mas, se essa abordagem for viável, o programa que joga Xadrez poderá, em princípio, aprender qualquer jogo, tornando-se um resolvidor de problemas genéricos.

Esta abordagem de “aprendizagem a partir do zero” foi aplicada através de uma arquitetura de processamento baseada em “Esquemas de Transição” (ver Capítulo 5) e os resultados foram promissores para resolução de problemas de baixa complexidade como, por exemplo, o do “Jogo da Velha”, sendo que, como esperávamos, uma mesma estrutura de processamento mostrou-se eficaz na solução de diversos outros tipos de problemas com essa mesma complexidade.

Neste ponto, voltamos a salientar que não estamos preocupados com a real compreensão que o agente irá ter, por exemplo, do “Jogo de Xadrez”. O que consideramos importante é a capacidade de um agente de aprender a “jogar” de forma semelhante a um especialista humano e obter um desempenho similar a este.

Escalas de Medição dos Graus de Adaptabilidade e Inteligência de Agentes Cognitivos

3.1 Introdução

Uma das primeiras propostas para verificar se uma máquina apresenta um comportamento “inteligente” foi formulada por Alan Turing [2] na década de 50, estabelecendo que:

“Se um computador pode realizar uma atividade de tal maneira que um especialista não consiga distinguir o seu desempenho do desempenho de um ser humano com determinada habilidade cognitiva, isto significa que o computador também possui essa habilidade cognitiva”.

O teste que ficou popularmente conhecido por “Teste de Turing” é uma particularização da proposição acima na qual um operador humano conversa com outro (ou com um computador) através de um terminal e tenta identificar se este é uma pessoa ou uma máquina. Muitos pesquisadores têm elaborado experimentos em que o teste de Turing tem sido utilizado para avaliar as técnicas mais recentes de programação. No entanto, até o presente momento, nenhum programa pôde ser aprovado no referido teste (na realidade muitos seres humanos também já foram reprovados). Para aplicações bastante restritas como, por exemplo, sistemas de reservas de passagens, os programas podem ter um desempenho bom, mas isso se deve ao fato de que, nesses casos, praticamente, todas as situações podem ser definidas em um conjunto limitado de regras.

A partir do teste de Turing, uma grande quantidade de testes específicos foram desenvolvidos visando não só avaliar a capacidade cognitiva de sistemas artificiais, mas também auxiliar o desenvolvimento desses sistemas. Dessa forma, o que encontramos, freqüentemente na literatura, são classes de problemas específicos (problema do caixeiro viajante, problema do caça-níqueis, etc.) e diversos tipos de jogos que são utilizados para comparar novas técnicas com as já disponíveis. Muitos desses problemas são ainda

insolúveis ou estão apenas parcialmente resolvidos, servindo, dessa forma, de desafio para a criação de novas soluções. De maneira geral, pode-se considerar que cada nova proposta de algoritmo ou estrutura de processamento apresenta também um problema (um teste) que ela se propõe a resolver com vantagens, assim, o número de possibilidades de testes encontrados na literatura é muito grande.

É importante observar que também, nas áreas das ciências biológicas, uma série de testes têm sido desenvolvidos visando avaliar a capacidade cognitiva de animais, principalmente de ratos, cães, macacos e golfinhos. Esses testes vão de atividades simples, como apertar uma alavanca para obter comida, a ações muito mais complexas (incluindo até capacidades de comunicação) nas quais, por vezes, o desempenho dos animais surpreende os próprios pesquisadores.

Dentro das ciências humanas, também foi proposta uma série de testes para avaliação de capacidades cognitivas, que vão do polêmico teste de QI a testes específicos para identificação de distúrbios de personalidade e problemas no desenvolvimento mental.

No âmbito da IA, observamos que o tipo de teste proposto por Turing é muito complexo quando a atividade a ser realizada é “conversar”, mas torna-se bem mais simples quando a atividade é, por exemplo, jogar Xadrez. Isso nos leva a alguns questionamentos:

- serão os sistemas que passam nestes testes realmente providos de “inteligência”?
- sistemas mais simples que certamente falhariam nestes testes seriam realmente desprovidos de qualquer grau de “inteligência”?
- somente sistemas muito complexos podem ser “inteligentes”?

De fato, observamos muitos casos em que o teste proposto por Turing leva a resultados contraditórios como, por exemplo, o fato do campeão mundial de Xadrez, Gary Kasparov perder para a máquina *Deep Blue* e, também, o fato de que muitos seres humanos já foram reprovados no teste de Turing.

Um dos grandes problemas da abordagem de Turing é que o fato de que um sistema apresentar um comportamento “inteligente” não implica que ele possua qualquer grau do tipo de “inteligência” que estamos buscando. Em primeiro lugar, para problemas com espaço de solução limitado, máquinas, como o *Deep Blue* (que trabalha apenas um

algoritmo de busca associado a uma capacidade maciça de processamento), podem resolver o problema sem ter um mínimo grau de “inteligência”. Além disso, sempre que um ser humano programa um computador, ele está de certa forma transferindo parte de sua “inteligência” à máquina para a resolução de um problema específico.

3.2 Escalas Propostas

No nosso entendimento, uma máquina efetivamente “inteligente” deve, em primeiro lugar, ter a capacidade de aprender a realizar uma atividade de maneira similar a um ser humano que aprende essa mesma atividade. Dessa forma, se o *Deep Blue* tivesse aprendido a jogar Xadrez, ele também poderia aprender a jogar qualquer outro jogo de tabuleiro, o que é uma característica natural de qualquer jogador “inteligente”. Assim, a proposta de Turing poderia ser reformulada para:

*Se um computador pode **aprender a** realizar uma atividade de tal maneira que um especialista não consegue distinguir o seu desempenho do desempenho de um ser humano com determinada habilidade cognitiva, isto significa que o computador também possui essa habilidade cognitiva.*

Um outro problema da abordagem de Turing é a divisão dos sistemas em apenas duas categorias (“inteligente” e “não inteligente”). Isso é muito restritivo, pois, como os testes aplicados são geralmente muito difíceis (como é o caso da conversação em linguagem natural), somente sistemas muito complexos teriam chance de passar, já que sistemas mais simples, ou seja, que possuíssem um certo grau de “inteligência”, seriam rotulados como “não inteligentes”.

Quando avaliamos a forma como os sistemas aprendem a executar algumas tarefas, observamos que, mesmo para aprender tarefas muito simples, um certo grau de “inteligência” é necessário e que, para tarefas mais complexas, esse grau deve ser maior.

Se olharmos novamente para as ciências naturais, veremos que os biólogos, ao estudar o comportamento de animais, afirmam muitas vezes que os animais possuem um certo grau de “inteligência” ou que um animal é mais “inteligente” do que outro (por exemplo, que um chimpanzé é mais inteligente do que um gorila). Essas afirmações são baseadas em testes realizados com animais, que podem ir de tarefas muito simples a tarefas bastante complexas. Dessa forma, se compararmos um conjunto de animais como, por

exemplo, um rato, um cachorro e um macaco, não teremos dificuldade em verificar que tarefas bem simples (como apertar um botão para conseguir comida) serão desempenhadas pelos três; tarefas um pouco mais complexas não serão desempenhadas pelo rato; por fim, certas tarefas só serão desempenhadas pelo macaco. Dessa forma, poderíamos afirmar que o cachorro é mais “inteligente” do que o rato e que o macaco é mais “inteligente” do que o cachorro.

Neste contexto, formulamos a hipótese de que é possível definir escalas que permitam medir o grau de “inteligência” de qualquer agente cognitivo, independentemente de sua complexidade e de sua natureza (artificial ou natural), de forma a estabelecer com uma certa precisão um grau numérico para cada agente como o da escala hipotética mostrado na Tabela 2. Geralmente, o que essa tabela nos indica gera muita polêmica, pois considera o ser humano como sendo um agente cognitivo e admite a possibilidade de que animais e sistemas artificiais possam ser “inteligentes”.

Uma escala deste tipo seria definida por uma série de testes de complexidade crescente com tarefas que o agente aprende a realizar dentro do mundo em que está inserido, sendo o grau de “inteligência” de um agente dado pelo índice da tarefa de maior complexidade que ele consegue realizar.

TABELA 2
ESCALA HIPOTÉTICA DE AVALIAÇÃO DE
GRAUS DE INTELIGÊNCIA DE AGENTES COGNITIVOS

Agente Cognitivo	Grau de “Inteligência”
Rato	4,5
Cachorro	18,7
Golfinho	45,2
Sistema X	12,3
Sistema Y	20,5
Homem	112,9

É importante observar que este tipo de escala é aplicável em âmbito bastante restrito de um agente cognitivo inserido em um mundo onde ele deve **aprender** a realizar uma atividade ou resolver um problema, sendo treinado para tal com base em estímulos de recompensa e punição. Para agentes cognitivos artificiais, o problema não é, em princípio, conhecido pelo ser humano que o “programou”. Dessa forma, mesmo tarefas muito simples, como apertar um botão para obter “comida”, poderão requerer estratégias

elaboradas, pois o programador do agente não sabe *a priori* que esta tarefa específica será imposta ao agente. Dessa maneira, regras do tipo “Se luz1 = ligada então pressione botão2”, não poderão ser pré-programadas, pois o programador deverá no máximo saber que, no ambiente, irão existir “luzes” e “botões”.

Um aspecto importante do presente trabalho não é a definição de novos testes, mas sim a proposta de agruparmos os milhares de testes já disponíveis (aplicáveis a animais, a computadores e a seres humanos) em escalas de medição de “inteligência” que permitam a definição de índices numéricos (ou graus de “inteligência”) válidos no contexto de cada escala.

Pelas pesquisas realizadas, observamos que escalas deste tipo não estão disponíveis e que os testes mais difundidos, como o de Turing, são compostos de tarefas muito complexas que não servem para avaliação de sistemas mais simples. Assim, isto se deve não por uma dificuldade técnica em idealizar e implementar o tipo de escala que estamos propondo, mas sim, pelo fato da aplicação do termo “inteligência” fora do contexto humano ser ainda muito criticado pela comunidade científica em geral. De fato, a afirmação de que a “inteligência” é a característica que nos distingue dos animais fica um pouco abalada no momento em que passamos a atribuir certos graus de inteligência a ratos, macacos e golfinhos.

Temos a convicção de que o tipo de escala de medição de “inteligência” que estamos propondo será uma ferramenta imprescindível tanto para avaliar como para aprimorar novos modelos, arquiteturas ou algoritmos de controle de agentes cognitivos. Em uma simples analogia, consideramos que estudar e desenvolver sistemas de “inteligência” artificial sem uma escala que permita medir o grau de “inteligência” de um dado sistema equivale a querer estudar terremotos sem ter uma escala para medir sua intensidade ou ainda estudar mineralogia sem uma escala para medição da dureza das rochas.

Se desejamos desenvolver sistemas artificiais efetivamente “inteligentes”, um dos primeiros passos é definir escalas que sejam bastante completas e também amplamente divulgadas e que permitam medir graus de “inteligência” em quaisquer agentes cognitivos, sejam eles animais, computadores ou seres humanos.

Em nossa pesquisa sobre agentes cognitivos artificiais e também avaliando experimentos realizados com animais, observamos dois grandes grupos de atividades que

demandam habilidades cognitivas distintas:

- atividades que são realizadas sem troca de informações simbólicas;
- atividades que dependam fundamentalmente de troca de informações simbólicas.

É conveniente mencionar que mesmo atividades do primeiro grupo podem ser bastante complexas, como é o caso, por exemplo, do “Jogo de Xadrez”, em que nenhuma comunicação entre os jogadores, em princípio, é necessária para o desenrolar do jogo. Porém, geralmente, essas tarefas mais complexas são mais rapidamente e facilmente assimiladas quando existe um processo de comunicação entre os agentes.

Desta forma, consideramos a necessidade de definir inicialmente duas escalas para a medição do grau de “inteligência” [32], para cada tipo de atividade acima citada:

- grau de adaptabilidade (GA);
- grau de inteligência simbólica (GIS).

Estas escalas serão descritas nas próximas seções.

3.3 Medição do Grau de Adaptabilidade

Antes de conceituar o parâmetro definido por grau de adaptabilidade (GA), devemos fazer algumas considerações sobre a capacidade de um agente cognitivo “viver” em um dado ambiente. O sentido que daremos ao termo “viver” é a necessidade de um agente realizar um certo conjunto de atividades para determinadas situações. O agente deve realizar ações que promovam estímulos positivos e não realizar aquelas que gerem estímulos negativos. A avaliação pretendida está inerentemente ligada ao meio em que o agente está inserido. Por exemplo, considerando-se um agente que dispõe de “maçãs” verdes e vermelhas e, ao ingeri-las, recebe um estímulo negativo e um positivo, respectivamente, podemos dizer, então, que o agente irá “viver” no ambiente se ele aprender a comer somente “maçãs” vermelhas, o que pode ser constatado se o agente comer um número significativo de “maçãs” vermelhas sem comer nenhuma verde.

Se o ambiente for, por exemplo, um tabuleiro do “Jogo da Velha”, o conceito “viver” pode estar associado a não perder nenhuma partida, e assim por diante.

O desempenho de um agente nas atividades realizadas sem troca de informações

simbólicas pode ser caracterizado por:

- desempenho satisfatório: o agente consegue “viver” em um dado ambiente;
- desempenho insatisfatório: o agente não consegue “viver” em um dado ambiente.

O desempenho do agente irá depender não somente de sua capacidade “mental”, mas também de suas capacidades sensoriais e motoras. Por exemplo, um agente que, para “viver” em um dado ambiente, necessite diferenciar cores de objetos, irá ter um desempenho insatisfatório caso sua visão seja monocromática. Da mesma forma, se ele necessitar levantar pesos de até 20 kg e possuir capacidade para apenas 10 kg, ele também terá desempenho insatisfatório.

Assim, em vez de usarmos o conceito de grau de desempenho, optamos por avaliar o agente segundo o seu grau de adaptabilidade, que estaria inerentemente ligado ao seu sistema de controle (sua capacidade de aprendizado) e seria independente de suas limitações sensoriais e motoras.

Definição 1 - Grau de Adaptabilidade do agente em relação ao meio: dados dois agentes A_1 e A_2 com capacidades sensoriais e motoras equivalentes, inseridos em um mesmo ambiente M e apresentando diferentes graus de adaptabilidade, o grau relativo de adaptabilidade do agente A_1 ao meio M será maior do que o do agente A_2 , se o desempenho do agente A_1 for satisfatório e o do agente A_2 for insatisfatório.

Assim, podemos escrever a seguinte equação:

$$G_a|_M(A_1) > G_a|_M(A_2) \quad (3.1)$$

Obs.: Se os dois agentes tiverem o mesmo desempenho, nada podemos afirmar sobre seus graus de adaptabilidade.

Definição 2 - Grau de complexidade do meio em relação ao agente: dados um agente A e dois ambientes M_1 e M_2 , considerando conjuntos equivalentes de informações sensoriais, porém graus de complexidade diferentes e confrontando-se o agente A com os ambientes M_1 e M_2 , o grau de complexidade de M_1 será menor do que

o de M_2 , se o agente tiver desempenho satisfatório no meio M_1 e insatisfatório no meio M_2 . Assim tem-se:

$$G_c|_A(M_1) > G_c|_A(M_2) \quad (3.2)$$

Obs.: Se o agente tiver o mesmo desempenho nos dois ambientes, nada podemos afirmar sobre o grau de complexidade dos ambientes.

Hipótese 1 - Dados um conjunto de ambientes (M_1, M_2, \dots, M_n) com diferentes graus de complexidade e um conjunto de agentes (A_1, A_2, \dots, A_m) com diferentes graus de adaptabilidade, os graus de complexidade relativos dos ambientes poderão ser obtidos, ordenando-se os ambientes, segundo o grau crescente (ou decrescente) de complexidade, que independe dos graus de adaptabilidade dos agentes. Assim:

$$G_c(M_i) < G_c(M_j) < \dots < G_c(M_k) < G_c(M_l) \quad (3.3)$$

Demonstração: Tomando-se dois meios quaisquer M_i e M_j , que têm por hipótese diferentes graus de complexidade, para um número tão grande quanto se deseje de agentes com diferentes graus de adaptabilidade, iremos identificar ao menos um agente que tenha desempenho satisfatório em um dos meios e insatisfatório no outro. Seguindo esse procedimento para todos os ambientes, poderemos ordená-los segundo um grau crescente (ou decrescente) de complexidade.

Definição 3 - Escala para medição do grau de adaptabilidade de agentes cognitivos: dado um conjunto de ambientes (M_1, M_2, \dots, M_n) com graus de complexidade crescente $(G_c(M_1) < G_c(M_2) < \dots < G_c(M_n))$, o grau de adaptabilidade de um dado agente será igual ao maior índice relativo ao conjunto de meios para o qual o agente apresentar desempenho satisfatório.

Obs.: O agente deve passar por todos os ambientes sem ter sua programação básica alterada. Para sistemas de IA, os responsáveis pela programação dos agentes não devem utilizar informações específicas de cada ambiente para programar as ações do

agente.

Como podem ser definidos diversos conjuntos de ambientes (múltiplas escalas), sugerimos que cada escala proposta receba uma denominação específica, indicando o máximo índice associado a ela.

3.4 Exemplo de Escala para Medição do Grau de Adaptabilidade

Uma escala para medição do grau de adaptabilidade de um agente cognitivo nos informa sobre a sua capacidade de realizar uma série de tarefas em cenários que são cada vez mais complexos. Esse tipo de escala é composto de níveis que representam tarefas específicas. Cada nível pode conter diversos subníveis, que indicam um aumento de complexidade dentro de um mesmo tipo de problema.

A escala “GA - Objeto Ação (9)” recebeu esta denominação por ter nove níveis que contêm tarefas relacionadas à apresentação de objetos e à realização de ações por parte do agente. Essa escala pode ser aplicada tanto a agentes cognitivos artificiais como a agentes naturais. A aplicação dessa escala em seres humanos só faz sentido como forma de validação da escala, pois em um contexto humano ela é tão simples que pode ser resolvida por qualquer ser humano.

Para o caso de sistemas artificiais, considera-se que o programador do agente deve desconhecer os detalhes específicos de cada problema. Dessa forma, a fim de garantir essa premissa, as tarefas poderiam ser ligeiramente modificadas, mantendo-se o mesmo grau de dificuldade. Assim, se desejarmos aplicar, por exemplo, ao problema do “Jogo da Velha”, poderemos modificar as regras para vitória e até mesmo o número de casas disponíveis. Assim, o sistema artificial deverá aprender as novas regras que não poderão ser prefixadas por um programador humano.

Um outro aspecto importante da escala proposta é o que diz respeito às interfaces de entrada e saída dos agentes artificiais. Mesmo um teste muito simples, como apertar um botão quando uma luz acende, requer um sistema sensorial e motor relativamente sofisticado para que o agente “veja” a luz, “veja” o botão e “observe” a posição de seu “braço” que deverá apertar o botão. Podemos sempre subdividir este problema, imaginando que um sistema de tratamento de imagens seria utilizado para ativar uma entrada binária quando a luz for acesa e um sistema de controle do “braço” do agente que

receberia um sinal binário indicando que o botão deve ser pressionado. Assim, sem perda de generalidade, podemos definir um agente muito simples, que possua como sensor apenas uma entrada digital e como atuador uma saída digital. Dessa forma, afastamo-nos das complexidades inerentes aos sensores e atuadores (que poderiam ser implementados com técnicas tradicionais de controle e de processamento de imagens) e nos concentramos na tarefa que queremos que o agente realize. Essa abordagem leva ao problema de que os sensores de entrada e saída do agente devam ser reconfigurados para determinadas tarefas, ou seja, que o agente seja reprogramado para saber que essas entradas e saídas existem, o que irá interferir no processo contínuo de aprendizagem que esperamos do agente.

Desta forma, poderíamos definir uma nova escala na qual o agente possuísse câmeras de vídeo como sensores e braços robotizados como atuadores. Nesse caso, mesmo para tarefas muito simples, os agentes já seriam relativamente complexos.

Além dos estímulos de reforço positivo e negativo, durante o processo de treinamento, um “mentor” humano poderá assumir momentaneamente o controle das saídas do agente, a fim de demonstrar qual a tarefa a ser realizada (treinamento supervisionado).

3.4.1 Descrição da Escala

Na escala GA-objeto ação, o ambiente em que o agente está inserido é composto de um cenário, no qual objetos são apresentados e algumas ações podem ser executadas pelo agente. Os objetos possuem atributos percebidos pelo agente, como sua forma e sua cor, e o agente recebe do ambiente reforços positivos e negativos em resposta às suas ações. O grau de complexidade do ambiente varia de acordo com o número de objetos, sua variedade de tipos e a quantidade de comandos disponíveis para o agente.

Os objetos mais simples que podemos apresentar a um agente são dados por uma entrada binária, que pode ser uma “luz” sendo acesa diante de um animal, ou uma entrada digital em um sistema artificial. A ação mais simples que um agente pode realizar é representada por um botão (ou uma alavanca) que deve ser pressionado, sendo modelado como uma saída digital. Na medida em que as tarefas evoluem, objetos mais complexos podem ser utilizados, como os padrões binários apresentados na Fig. 3. Ações mais complexas podem ser obtidas pela utilização de diversas saídas. Dessa forma, se

desejarmos que o agente “desenhe” um dos objetos representados na Fig. 3, basta disponibilizarmos 9 saídas digitais.

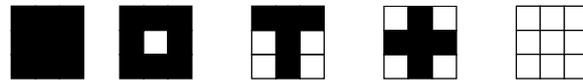


Fig. 3. Exemplo de objetos.

A seguir, uma breve descrição de cada nível da escala é apresentada.

Nível 1

Ambiente: O agente possui somente entradas de reforço positivo ou negativo, e pode pressionar um botão.

Tarefa a ser realizada: O agente deve aprender a pressionar o botão.

Estímulos aplicados: Quando o botão é pressionado, um reforço positivo é gerado.

Um tempo muito longo sem pressionar o botão gera um reforço negativo.

Subnível 1.2

Tarefa a ser realizada: O agente deve aprender a pressionar o botão duas vezes consecutivas, realizar uma pausa e repetir esse ciclo.

Subnível 1.3

Tarefa a ser realizada: O agente deve aprender a pressionar o botão N vezes consecutivas, realizar uma pausa e repetir esse ciclo.

Nível 2

Ambiente: Além das entradas de reforço, o agente pode “ver” um objeto (por exemplo, uma luz verde sendo ligada) e pode pressionar um botão.

Tarefa a ser realizada: Quando o objeto é apresentado, o botão deve ser pressionado.

Estímulos aplicados: Realizar a tarefa gera um reforço positivo. Apertar o botão sem objeto ou não apertá-lo algum tempo após a apresentação do objeto gera um reforço negativo.

Subnível 2.2

Tarefa a ser realizada: O agente deve aprender a pressionar o botão duas vezes ao “ver” o objeto.

Subnível 2.3

Tarefa a ser realizada: O agente deve aprender a pressionar o botão N vezes ao “ver” o objeto.

Nível 3

Ambiente: O agente pode “ver” dois tipos de objetos (por exemplo, uma luz verde e outra vermelha) e pode pressionar um botão.

Tarefa a ser realizada: O botão deve ser pressionado apenas para um tipo de objeto.

Estímulos aplicados: Realizar a tarefa gera um reforço positivo. Apertar o botão sem objeto ou para o objeto errado, ou ainda não apertá-lo algum tempo após a apresentação do objeto correto gera um reforço negativo.

Obs.: A partir deste nível, os estímulos aplicados são bastante similares, ou seja, estímulos positivos são aplicados quando a tarefa é corretamente realizada e, em caso contrário, são aplicados estímulos negativos. Dessa forma, nos próximos níveis, esses estímulos não serão mais descritos.

Subnível 3.2

Tarefa a ser realizada: O agente deve aprender a pressionar o botão duas vezes ao “ver” o objeto correto.

Subnível 3.3

Tarefa a ser realizada: O agente deve aprender a pressionar o botão N vezes ao “ver” o objeto.

Nível 4

Ambiente: O agente pode “ver” objetos (um de cada vez) com várias formas (por exemplo, os objetos da Fig. 3). O agente pode pressionar dois botões: um gera a ação de “comer” o objeto e o outro “pega” um novo objeto.

Tarefa a ser realizada: O agente deve procurar novos objetos “comendo” apenas alguns tipos de objeto (por exemplo, bolas e quadrados).

Subnível 4.2

Ambiente: Além de objetos de várias formas, são apresentadas outras características que são identificadas pelo agente, como, por exemplo, objetos de várias

cores.

Subnível 4.3

Ambiente: As formas dos objetos são distorcidas pela inclusão de um baixo nível de ruído, como mostrado na Fig. 4.



Fig. 4. Exemplos de objetos distorcidos por ruído.

Nível 5

Ambiente: O agente pode “ver” objetos de várias formas e cores, sendo apresentados até dois objetos simultâneos (um na posição *A* e outro na posição *B*). O agente pode pressionar três botões que agem somente sobre os objetos apresentados na posição *A*: o primeiro gera a ação de “comer” o objeto; o segundo “pega” um novo objeto (de uma fila de objetos ocultos); e o terceiro retorna a um objeto anterior (que não foi comido).

Tarefa a ser realizada: Um objeto é apresentado na posição *B*. O agente deve procurar um objeto igual na posição *A* e “comer” esse objeto.

Subnível 5.2

Ambiente: As formas dos objetos são distorcidas pela inclusão de um baixo nível de ruído, como mostrado na Fig. 4.

Nível 6

Ambiente: O agente pode “ver” objetos de várias formas e cores, sendo vários objetos apresentados em um “tabuleiro” e um objeto apresentado fora dele. Além disso, o agente pode “ver” um cursor sobre uma das casas do tabuleiro. O agente pode pressionar seis botões: o primeiro gera a ação de “colocar” o objeto que está fora para dentro do tabuleiro na posição onde o cursor está posicionado (se esta casa estiver vazia); o segundo e o terceiro botões “pegam” novos objetos de uma fila de objetos ocultos; o quarto e o quinto botões movem o cursor pelas casas do tabuleiro (como se elas estivessem numeradas de forma seqüencial); e o sexto botão move um objeto para fora do tabuleiro (se o cursor estiver sobre um objeto), ocultando o objeto que ocupava essa posição.

Tarefa a ser realizada: Um objeto é apresentado em uma determinada posição do tabuleiro. O agente deve levar o cursor até essa posição e tirar o objeto do tabuleiro.

Subnível 6.2

Tarefa a ser realizada: Um objeto é apresentado em uma determinada posição do tabuleiro. O agente deve tirar o objeto do tabuleiro e colocá-lo de volta na mesma posição.

Subnível 6.3

Tarefa a ser realizada: Vários objetos são apresentados no tabuleiro. O agente deve tirar todos eles do tabuleiro e, a seguir, recolocá-los de volta.

Nível 7

Ambiente: O mesmo do nível 6.

Tarefa a ser realizada: São apresentados alguns objetos com características em comum (por exemplo, da mesma cor), devendo o agente colocar no tabuleiro outros objetos desse tipo.

Subnível 7.2

Tarefa a ser realizada: Uma seqüência incompleta de objetos é apresentada, devendo o agente completar a seqüência.

Nível 8

Ambiente: O mesmo do nível 6, com o tabuleiro e os objetos definidos de forma a permitir a execução de jogos com apenas dois tipos de peças.

Tarefa a ser realizada: O agente deve ser capaz de vencer (ou empatar) um jogo de complexidade equivalente ao “Jogo da Velha”.

Subnível 8.2

Tarefa a ser realizada: O agente deve ser capaz de vencer (ou empatar) um jogo de complexidade equivalente ao “Jogo de Damas”.

Nível 9

Ambiente: O mesmo do nível 6, com o tabuleiro e os objetos definidos de forma a permitir a execução de jogos com vários tipos de peças .

Tarefa a ser realizada: O agente deve ser capaz de movimentar as peças e realizar uma partida de um jogo com complexidade equivalente ao “Jogo de Xadrez”.

Subnível 9.2

Tarefa a ser realizada: O agente deve ser capaz de vencer um jogador iniciante.

Subnível 9.3

Tarefa a ser realizada: O agente deve ser capaz de vencer um jogador experiente.

3.5 Medição do Grau de Inteligência Simbólica

Antes de se definir este parâmetro, devemos fazer algumas considerações sobre a capacidade de manipulação de símbolos ou a capacidade de comunicação de um agente cognitivo.

Para um dado ambiente, podemos definir um conjunto limitado de símbolos que expressem conceitos distintos e que estejam associados a elementos (concretos e/ou abstratos) desse ambiente. Esse conjunto de símbolos é denominado “vocabulário coerente” associado ao meio. Essa limitação deve-se ao fato de o agente, em princípio, não conhecer nenhum símbolo e ter de aprender o seu significado por intermédio de outros agentes inseridos no meio. Por exemplo, para um ambiente em que são representados apenas objetos em preto e branco, o símbolo “vermelho” (correspondente à cor vermelha) não faz parte do vocabulário coerente do meio e, portanto, não poderia ser ensinado ao agente. Os primeiros símbolos transmitidos (ensinados) estão geralmente associados a objetos e ações concretas, em seguida poderão ser ensinados símbolos relacionados a conceitos mais abstratos.

A capacidade de comunicação de um dado agente pode ser medida pela forma com que ele domina o vocabulário coerente definido para o meio. Esse domínio pode ser verificado de duas maneiras:

- solicitando ao agente que realize uma determinada ação (por exemplo: “Pegue bola verde”);
- solicitando ao agente que responda a uma questão (por exemplo: “Cor bola?”).

Se o agente for capaz de realizar as tarefas e responder corretamente as questões formuladas, podemos concluir que ele tem domínio sobre um dado vocabulário coerente e consegue se comunicar através desse vocabulário.

Para meios mais complexos, não só a capacidade de comunicação do agente estará

sendo medida, como também a sua capacidade de responder a questões genéricas e de realizar associações lógicas e inferências.

Desta forma, pretendemos avaliar o agente segundo um “grau de inteligência simbólica”, que estaria relacionado à capacidade de comunicação do agente, como também ao conceito de inteligência, normalmente associado aos seres humanos.

Definição 4 - Grau de inteligência simbólica do agente em relação ao ambiente: Dados dois agentes A_1 e A_2 com capacidades sensoriais e motoras equivalentes, diferentes graus de inteligência, confrontados com um mesmo ambiente M para o qual um dado vocabulário coerente é definido, o grau de inteligência simbólica relativo ao meio M do agente A_1 será maior do que o do agente A_2 , se o agente A_1 dominar o vocabulário coerente e o agente A_2 não dominá-lo. Assim, tem-se:

$$G_{is}|_M(A_1) > G_{is}|_M(A_2) \quad (3.4)$$

Obs.: Se os dois agentes tiverem o mesmo desempenho, nada podemos afirmar sobre seus graus de inteligência simbólica.

Definição 5 - Grau de complexidade simbólica do meio em relação ao agente: Dados um agente A e dois ambientes M_1 e M_2 , com vocabulários coerente distintos (ou aplicáveis em diferentes contextos), confrontando o agente com os ambientes, o grau de complexidade simbólica de M_1 será menor do que o de M_2 , se o agente A dominar o vocabulário coerente de M_1 e não o de M_2 . Então:

$$G_{cs}|_A(M_1) > G_{cs}|_A(M_2) \quad (3.5)$$

Obs. 1: Se o agente tiver o mesmo desempenho nos dois ambientes, nada podemos afirmar sobre seus graus de complexidade simbólica.

Obs. 2: Se o vocabulário coerente de um meio estiver contido no do outro, o de menor vocabulário terá menor grau de complexidade simbólica.

Hipótese 2 - Dados um conjunto de ambientes (M_1, M_2, \dots, M_n) com diferentes

graus de complexidade simbólica e um conjunto de agentes (A_1, A_2, \dots, A_m) com diferentes graus de inteligência simbólica, os graus de complexidade simbólica relativos poderão ser obtidos ordenando-se os ambientes segundo o grau crescente (ou decrescente) de complexidade, que independe do grau de inteligência dos agentes. Assim:

$$G_{cs}(M_i) < G_{cs}(M_j) < \dots < G_{cs}(M_k) < G_{cs}(M_l) \quad (3.6)$$

Demonstração: Idêntica à da Hipótese 1.

Definição 6 - Escala para medição do grau de inteligência simbólica de agentes cognitivos: Dado um conjunto de ambientes (M_1, M_2, \dots, M_n) com graus de complexidade simbólica crescente $(G_{cs}(M_i) < G_{cs}(M_j) < \dots < G_{cs}(M_k) < G_{cs}(M_l))$, o grau de inteligência simbólica de um dado agente será igual ao maior índice relativo ao conjunto de meios para o qual o agente demonstrar domínio do vocabulário coerente definido para o meio.

Obs.: Em princípio, o agente não deve possuir qualquer domínio de vocabulário. No entanto, o vocabulário aprendido em um determinado nível poderá ser usado no próximo nível.

Como podem ser definidas diversas escalas, sugerimos que, na denominação de cada uma delas, seja indicado o máximo índice da escala.

3.6 Exemplo de Escala para Medição do Grau de Inteligência Simbólica

Uma escala para medição do grau de inteligência simbólica de um agente cognitivo nos informa sobre a sua capacidade de manipular símbolos e de se comunicar. Esse tipo de escala é composto de uma série de situações (definidas por vários níveis) em que conceitos cada vez mais complexos são apresentados ao agente.

A escala “GIS – ABCE (11)” recebeu esta denominação por ter 11 níveis e utilizar palavras compostas com as letras “A”, “B”, “C” e “E”.

De forma similar à da escala anterior, as interfaces de entrada e saída dos agentes artificiais são bastante simples, compondo-se apenas de entradas e saídas binárias. As palavras são “lidas” pelo agente através de uma seqüência de letras, sendo adotadas sempre

palavras de quatro letras. Da mesma forma, o agente pode “escrever” selecionando uma letra de cada vez. As letras escritas pelo agente também são lidas por ele, simulando um processo de “ouvir o que se está falando”.

Além dos estímulos de reforço positivo e negativo, durante o processo de treinamento, um “mentor” humano pode assumir momentaneamente o controle das saídas do agente (inclusive as de escrita), a fim de efetivar um treinamento supervisionado.

3.6.1 Descrição da Escala

Na escala “GIS – ABCE (11)”, o ambiente em que o agente está inserido é composto de um cenário em que, além de objetos, símbolos são apresentados ao agente. Utilizando apenas quatro letras formamos uma série de símbolos que compõem um vocabulário coerente, conforme mostrado na Tabela 3. Através de vários “botões”, o agente pode realizar uma série de ações, inclusive escrever seqüências de símbolos.

A seguir, uma breve descrição de cada nível da escala é apresentada.

TABELA 3
EXEMPLOS DE VOCABULÁRIO COERENTE

Símbolo	Significado
BEBE	Comer objeto
BECA	Novo objeto
BECE	Objeto anterior
BACA	Quadrado
BACE	Círculo
BABE	Cruz
BABA	Triângulo
CECA	Preto
CECE	Azul
CEBE	Vermelho
CEBA	Verde
CABA	Cor
CACE	Forma
CABE	Ação
CACA	Objeto
ACAC	Qual - Igual a -?
ABAC	Sim- Verdadeiro
ECEB	Não - Falso
ABEB	Bom - Prazer
ECEC	Mau - Dor

Nível 1

Ambiente: O agente pode “ver” objetos de várias formas e cores, sendo apresentado um objeto de cada vez. O agente pode pressionar três botões que agem sobre os objetos: o primeiro gera a ação de “comer” o objeto; o segundo “pega” um novo objeto (de uma fila de objetos ocultos); e o terceiro “retorna” um objeto anterior (que não foi comido). O agente possui também quatro botões de “escrita” que selecionam as quatro letras e permitem a formação de palavras.

Obs.: Será utilizado, nos próximos níveis, este mesmo ambiente.

Tarefa a ser realizada: O agente deve repetir as palavras que recebe.

Nível 2

Tarefa a ser realizada: Símbolos que representam as ações dos três primeiros botões (BEBE, BECA e BECE) são apresentados sempre que o agente efetua uma delas. Após um período de treinamento, os símbolos relativos às ações são apresentados, devendo o agente realizá-las.

Nível 3

Tarefa a ser realizada: Os botões de ação são desabilitados e o agente deve realizá-las escrevendo o nome das mesmas.

Nível 4

Tarefa a ser realizada: Palavras que representam as formas dos objetos (BACA, BACE, BABE e BABA) são apresentadas em associação aos objetos. Após um período de treinamento, é apresentado o símbolo de determinada forma, devendo o agente procurar um objeto com esta forma.

Nível 5

Tarefa a ser realizada: Palavras que representam as cores dos objetos (CECA, CECE, CEBE e CEBA) são apresentadas em associação aos objetos. Após um período de treinamento, é apresentada uma frase contendo uma ação e uma determinada característica, devendo o agente realizar essa ação (por exemplo, BEBE BABA CECE → comer triângulo azul).

Nível 6

Tarefa a ser realizada: O conceito de pergunta (ACAC) é apresentado. Após apresentar um objeto e o símbolo ACAC, o agente deverá responder com uma palavra relacionada ao objeto e não com uma ação. Por exemplo, BEBE BABA CECE ACAC equivale à pergunta “comeu um triângulo azul?”.

Nível 7

Tarefa a ser realizada: Os conceitos de forma (CACE) e cor (CABA) são apresentados. Após um período de treinamento, são apresentados um objeto e a frase “CACE ACAC” (qual a forma?) ou a frase “CABA ACAC” (qual a cor?), devendo o agente responder adequadamente.

Nível 8

Tarefa a ser realizada: Os conceitos “sim” (ABAC) e “não” (ECEB) são apresentados. Após um período de treinamento, é apresentado, por exemplo, um objeto azul e a frase “CABA CECE ACAC “ (cor azul?), devendo o agente responder “ABAC”.

Nível 9

Tarefa a ser realizada: Os conceitos “prazer” (ABEB) e “dor” (ECEC) são apresentados. Após um período de treinamento, é apresentada uma afirmação do tipo “BEBE BACE CECE ACAC” (comer círculo azul?), devendo o agente responder “ECEC”, no caso em que a ação de “comer” esse objeto causar “dor”, ou “ABEB”, no caso de causar prazer.

Nível 10

Tarefa a ser realizada: Utilizando todo o vocabulário, solicitar uma série de ações e uma série de perguntas do tipo “CABA BACE ACAC” (qual a cor do círculo?) ou “BEBE BABA CECA” (coma triângulo preto) de forma a validar o processo de comunicação.

Nível 11

Neste nível, esperaríamos um certo grau de independência do agente com o diálogo a seguir:

Mentor - “BEBE BABE CEBE”

Agente - “ECEB”

Mentor - “ACAC ECEB”

Agente - “BEBE BABE CEBE ACAC ECEC”¹

3.7 Considerações sobre as Escalas Propostas

As escalas para medição dos graus de adaptabilidade e inteligência simbólica apresentadas foram baseadas em testes que se enquadram com a proposta original de Turing sobre certos aspectos, porém se diferenciam em outros:

- a escala é aplicável somente a agentes cognitivos que inicialmente não possuem qualquer informação sobre o ambiente onde estão inseridos e devem aprender de forma contínua e interativa;
- em vez de trabalhar com apenas dois níveis: “inteligente” e “não inteligente”, definimos uma série de níveis que estão associados à resolução de problemas com graus de dificuldade crescentes;
- para agentes cognitivos artificiais, os programadores humanos do agente deverão conhecer apenas os tipos de problemas a serem resolvidos pelo agente, sem conhecer detalhes específicos, dessa forma eles deverão desenvolver um agente que efetivamente aprenda a resolver os problemas.

Assim como Turing, estamos interessados na “inteligência” em um sentido mais próximo ao aplicável aos seres humanos, principalmente no que diz respeito à resolução de problemas genéricos. Entretanto, as escalas propostas não levam em conta aspectos mais abstratos que algumas vezes estão associados à inteligência, tal como criatividade e autoconsciência, considerando apenas as ações e reações que possam ser efetivamente medidas.

3.8 Resultados Experimentais

As duas escalas propostas neste capítulo são apenas um esboço rudimentar do tipo

¹ Traduzindo livremente teremos algo como:

Mentor - “Coma cruz vermelha”

Agente - “Não”

Mentor - “Como Não?”

Agente - “Comer cruz vermelha causa dor”

de escala que consideramos exequível. Para uma validação preliminar dessas escalas, elas foram aplicadas a voluntários humanos. Todos os agentes cognitivos humanos conseguiram obter um desempenho máximo nas duas escalas, variando apenas no tempo de aprendizagem. Ficou demonstrado mesmo em um ambiente muito simples, através de tarefas com graus de dificuldade crescente, que uma série de informações poderiam ser passadas ao agente, inclusive uma nova linguagem simbólica que foi aprendida independentemente da linguagem normalmente utilizada pelo agente.

Além de validar as escalas, a observação da forma de interação dos agentes cognitivos humanos com o ambiente virtual revela um pouco sobre os processos de aprendizagem por eles utilizados que, em princípio, deveriam ser aplicados para treinar agentes artificiais que fossem resolver os mesmos tipos de problemas.

Além de testes com agentes humanos, realizamos testes com agentes artificiais implementados com estruturas de controle baseadas em esquemas de transição que serão apresentados no próximo capítulo. Os resultados dessas estruturas foram promissores quanto ao grau de adaptabilidade, sendo observado um grau 3,0 (três), na escala GA Objeto Ação (9). Porém, esses resultados foram muito pobres quanto ao grau de inteligência simbólica, obtendo apenas um grau 1.0 (um) na escala GIS ABCE (11).

A utilização destas escalas para testar as estruturas de esquemas de transição propostas levou a um processo de realimentação em que as estruturas foram avaliadas e as suas principais limitações foram identificadas, o que permitiu um aprimoramento dessas estruturas.

Para uma validação mais abrangente da metodologia proposta, torna-se necessário um minucioso processo de especificação e detalhamento das escalas, seguido de uma etapa experimental que levasse em conta diversos tipos de agentes artificiais e naturais.

3.9 Conclusões

Do nosso ponto de vista, a obtenção de sistemas efetivamente “inteligentes” somente será possível quando pudermos, de alguma forma, identificar os diversos graus de inteligência dos sistemas em desenvolvimento.

Desta forma, a primeira contribuição deste trabalho não são as escalas propostas, mas sim a metodologia apresentada para a definição dessas escalas. Um dos pontos

fundamentais dessa proposta é que o sistema aprenda do “zero”, de forma similar ao aprendizado humano. Para tornar bastante claro esse ponto, consideremos o último nível da escala GA Objeto Ação (9) que consiste no problema de jogar Xadrez. Apesar de existir uma infinidade de programas com excelente desempenho no “Jogo de Xadrez” (incluindo o exemplo já tão citado do *Deep Blue*), todos esses programas atingiriam apenas um grau mínimo na escala proposta. Isso se deve ao fato de que, na metodologia proposta, os programadores desses sistemas deveriam desconhecer as regras específicas do problema a ser tratado. Dessa forma, bastaria definir o “Jogo de Xadrez” com regras levemente alteradas, excluindo uma peça ou modificando sua forma de movimento ou até modificando o objetivo final do jogo (por exemplo, capturar a dama em vez do rei), obtendo, assim, um problema de complexidade equivalente ao Xadrez, que até mesmo o *Deep Blue* iria falhar completamente, pois todo o conhecimento que ele tem de Xadrez foi repassado através de regras formuladas por seus programadores e que iriam para o “lixo” em um contexto levemente diferente do original. Dessa forma, a única possibilidade de passar no teste do “Jogo de Xadrez” proposto é construir um programa que aprenda a jogar de forma similar ao aprendizado de um jogador humano. É fácil perceber que, se esse programa aprende a jogar Xadrez, ele poderá também adaptar-se a mudanças de regra e poderá, em princípio, aprender qualquer tipo de jogo de tabuleiro.

Escalas de medição semelhantes às propostas neste trabalho são de vital importância para o desenvolvimento da IA, mas só poderão ser concretizadas com o esforço conjunto de várias equipes de pesquisa. Essas escalas possibilitariam uma definição e a quantização de diversos aspectos relacionados com os termos “inteligência” e “comportamento inteligente”. Para serem efetivas, essas escalas deveriam ser mundialmente aceitas e, dessa forma, construí-las e divulgá-las é um objetivo que extrapola, em muito, o âmbito desta tese de doutorado.

Capítulo 4

Um Novo Paradigma de Controle Baseado em Esquemas de Transição Aplicado à Criação de Agentes Cognitivos

4.1 Introdução

Neste capítulo, abordaremos um novo modelo de controle de agentes cognitivos. O modelo apresentado não se preocupa com sinais de entrada e sinais de saída, mas trabalha com subespaços de causa e de efeito que podem conter sinais de entrada, sinais de saída ou ambos. Os modelos de controle tradicionais são baseados em seqüências de “entrada – processamento - saída” que, apesar de serem muito naturais para nós, também fecham nossos olhos para outros modelos de processamento possíveis.

Como já enfatizamos no primeiro capítulo, os conceitos de “entrada” e “saída” que permeiam todos os nossos modelos de controle não são tão elementares como normalmente pensamos.

No paradigma de controle tradicional, os sistemas são modelados em blocos, sendo o bloco mais simples composto de todas as entradas conectadas a uma caixa preta, que processa os sinais de entrada, definindo valores para os sinais de saída. Essa caixa preta pode ser decomposta em uma série de caixas menores, e assim sucessivamente. Cada caixa preta processa seu conjunto de dados de entrada e define seus dados de saída. Esse modelo é a base de todas as técnicas de programação estruturada.

Vamos mostrar, neste capítulo, que esta forma de operação não é a única possível e que existe um modelo em que os sinais são tratados como relações de causa efeito. Como não estamos tratando apenas de um novo conceito de processamento, mas sim de um novo paradigma de processamento, vamos apresentá-lo na seqüência de seu desenvolvimento. Iniciaremos por uma discussão sobre formas tradicionais de treinamento não supervisionado de agentes cognitivos. Em seguida, é apresentando o modelo de esquemas proposto por Piaget [24] e o modelo de descrição de processos proposto por Markov [39].

Antes de entrar nesta discussão, apresentaremos um exemplo no âmbito humano do que entendemos por processamento de objetivos, mostrando uma cadeia de processos que parte de um objetivo muito amplo que vai sendo desdobrado até que ações específicas sejam definidas.

4.2 Exemplo de Processamento de Objetivos

Um ponto importante do modelo proposto é que “objetivos maiores” são naturalmente desdobrados em “objetivos menores”, e assim sucessivamente, até que uma “ação específica” seja definida. É um fator primordial que esse modelo possa ser aplicado a um agente, em princípio, tão complexo como um ser humano, ilustrado pelo exemplo a seguir.

Consideremos um agente que recebe dois objetivos máximos dados por “sentir prazer” e “não sentir dor”. O objetivo “sentir prazer” pode ser desdobrado em um conjunto de objetivos complementares dado por [“comer se estiver com fome”; “beber água se estiver com sede” ; “dormir se estiver com sono”, “tomar banho se estiver necessitado”]. Em um dado momento, o objetivo “tomar banho” pode ser ativado e, estando este subordinado ao objetivo maior que é “não sentir dor” (que estará sempre ativo), podem ser ativados “objetivos negados” (ou anti-objetivos) do tipo [“não entrar em água muito fria”, “não entrar em água muito quente”]. Esses objetivos, por sua vez, podem ativar o objetivo “regular a temperatura do chuveiro”, que ativa objetivos como [“abrir mais a torneira se a temperatura estiver muito alta”, “fechar mais a torneira se a temperatura estiver muito baixa”].

Neste ponto, um problema como um defeito na resistência do chuveiro pode impedir que o objetivo “não entrar em água muito fria” seja atingido. Assim, o objetivo “substituir a resistência do chuveiro” pode ser ativado por um mecanismo de resolver conflitos gerados pelo “travamento” (impossibilidade de atingir) de um objetivo ativo. Se uma resistência nova não for encontrada no local, o objetivo “ir ao mercado comprar uma resistência de chuveiro” pode ser ativado. Esse objetivo, por sua vez, pode ativar outros objetivos secundários como “colocar uma roupa antes de sair de casa” ou “pegar dinheiro no caixa eletrônico”, etc. É interessante observar que sempre existem diversos caminhos para atingir um dado objetivo, assim, o objetivo “ir ao mercado” pode gerar um dos

seguintes objetivos: “ir ao mercado de carro”, “ir ao mercado de bicicleta”, “ir ao mercado de ônibus”. O objetivo “ir ao mercado de carro”, ao ser escolhido, gera uma cadeia de outros novos objetivos do tipo: “procurar a chave do carro”, “entrar no carro”, “ligar o carro”, “dirigir até o mercado” e “estacionar o carro”. Cada um desses objetivos é desdobrado em objetivos menores até o nível de ações individuais, por exemplo “ligar o carro” pode gerar os objetivos “botar a marcha em ponto neutro” e “girar a chave na ignição” que, por sua vez, geram as ações “colocar o pé esquerdo sobre a embreagem”, “pressionar o pé”, “colocar a mão direita aberta sobre o câmbio”, “fechar a mão direita”, “mover o braço direito até o câmbio ficar na posição neutra”, e assim sucessivamente.

É interessante observar, neste exemplo, como o macro objetivo “sentir prazer” pode ser desdobrado até uma ação muito específica como “colocar a mão direita aberta sobre o câmbio”. Apesar dos desdobramentos anteriormente apresentados carecerem de um formalismo matemático (que será considerado mais adiante), eles ilustram bem o poder do modelo de processamento de objetivos proposto neste trabalho. É bom lembrar que, a cada momento, existe uma série de objetivos ativos competindo por recursos, sendo que a maior parte deles fica em estado de latência. Cada um desses objetivos ativos tem um certo grau de prioridade, que pode ser elevado ou diminuído por fatores internos ou externos ao agente. Por exemplo, o objetivo “tomar água” pode ser priorizado quando o agente passar perto de um bebedouro, e diminuído quando o agente estiver assistindo a uma peça de teatro.

4.3 Treinamento de Agentes Autônomos

O treinamento condicionado (*reinforcement learning*) é um campo de pesquisa da área de Inteligência Artificial que vem crescendo significativamente nos últimos anos. O objetivo desse tipo de treinamento é a programação de agentes autônomos através de um processo de condicionamento controlado por estímulos de recompensa ou punição, sem que haja necessidade de programação específica.

A maioria dos trabalhos nesta área, tais como *Q-learn* [33], *adaptive heuristic critic* [34], *TD(λ)* [35], *associative reinforcement comparison* (ARC) [36], *complementary reinforcement backpropagation* (CRBP) [37], considera que as ações do agente atuam sobre o sistema a ser controlado, gerando sinais de reforço positivo ou

negativo. Esses sinais de reforço são utilizados para alimentar mecanismos de aprendizagem normalmente dissociados da função de controle do agente. Assim, um dado agente pode ser considerado como uma função complexa de suas saídas em função de suas entradas. Normalmente, essa função complexa vai sendo modelada aos poucos por um processo de treinamento baseado nos sinais de reforço. Podemos considerar também um modelo no qual, a cada instante, o agente observa o estado de todas as suas entradas e define o melhor valor para todas as suas saídas, que, nesta mesma condição, já tenham gerado um reforço positivo. O grande problema desse modelo é que todas as entradas e saídas são consideradas simultaneamente, sendo muito difícil subdividir a função de controle em subfunções organizadas hierarquicamente.

O modelo de processamento de objetivos proposto neste trabalho permite o controle de agentes, utilizando os sinais de reforço positivo e negativo em um enfoque diferente. Como o modelo prevê naturalmente a hierarquização de objetivos, os sinais de reforço irão definir os objetivos máximos do agente como sendo: receber reforços positivos (sentir prazer) e não receber reforços negativos (não sentir dor). Esse novo enfoque permite o treinamento de agentes em tarefas muito mais complexas do que as permitidas pelos modelos de treinamento condicionados tradicionais, pois, em vez de lidar com uma única função de controle do agente, utilizamos uma hierarquia de relações de causa e efeito que podem ser muito mais facilmente determinadas.

4.4 Treinamento com Reforço

De forma geral, percebe-se que todos os seres vivos são capazes de se adaptar a um meio ambiente, basicamente, através de dois tipos de comportamentos: instintivos e condicionados.

O treinamento condicionado é um processo em que um agente, inserido em um ambiente, deve descobrir soluções através da aprendizagem por tentativa e erro. O agente recebe uma série de informações do ambiente (inclusive os estímulos de reforço) e simultaneamente interage com o ambiente, como ilustrado na Fig. 5.

Existem, basicamente, dois tipos de reforço: reforço imediato e reforço com atraso. No reforço imediato, o estímulo (positivo ou negativo) é recebido após a realização de cada ação. No reforço com atraso, o estímulo é recebido após a realização de um

conjunto de ações ou após um certo tempo.

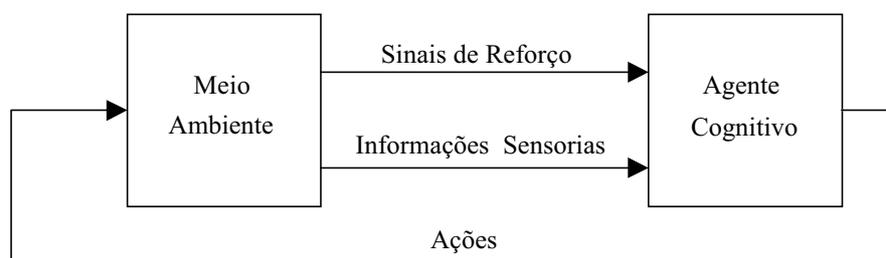


Fig. 5. Representação de um agente cognitivo inserido no meio ambiente.

Tradicionalmente, o agente é modelado através de uma função de controle que, para um dado estado de entrada, gera um ou mais estados de saída. Dessa forma, o agente deve realizar um mapeamento entrada-saída que leve em conta os estímulos de reforço e indique qual a ação de maior *ganho* (a que tenha a maior possibilidade de gerar um estímulo positivo ou evitar um negativo) para cada estado de entrada. Segundo Kaelbling [38], existem duas estratégias principais para a determinação desse mapeamento. A primeira consiste na realização de uma busca exaustiva no ambiente a fim de determinar as ações que apresentem melhor desempenho para cada estado de entrada. Essa busca pode ser realizada através de algoritmos clássicos de otimização, e de outros algoritmos como, por exemplo, algoritmos genéticos. A segunda estratégia passa pelo uso de técnicas estatísticas e por métodos que permitem estimar a importância de determinadas ações em função de estados do ambiente, que vão sendo gradativamente explorados pelo agente. No treinamento condicionado, a segunda estratégia é a mais interessante, pois, normalmente, o agente inserido no ambiente passa por um processo de aprendizagem contínuo, no qual não é possível realizar uma busca exaustiva de todos os estados de entrada.

O problema de treinamento condicionado de um agente pode ser descrito através do modelo MDP (*Markov decision process*) proposto por Markov como apresentado em [39] que consiste em:

- um conjunto de estados de entrada (S);
- um conjunto de ações (A);
- uma função de recompensa R onde: $S \times A \rightarrow \mathfrak{R}(S)$ (dado um estado atual S , aplicando a ação A , teremos um conjunto de probabilidades de obter

recompensas distintas, representado por $\mathfrak{R}(S)$);

- uma função de transição de estado T onde: $S \times A \rightarrow \Pi(S)$ (dado um estado atual S , aplicando a ação A , obtemos um conjunto de probabilidades de obtenção de novos estados $\Pi(S)$).

Obs.: Sem perda de generalidade, esta definição também pode ser aplicada a espaços contínuos.

Cada elemento de $\Pi(S)$ representa uma função de distribuição de probabilidade sobre o conjunto S (ou seja, $\Pi(S)$ mapeia a probabilidade de transição dos estados S). Pode-se escrever então $T(s, a, s')$ como uma função de probabilidade de transição do estado s para o s' quando se executa a ação a .

Um ambiente é dito MDP quando T não depende de qualquer estado ou ação anterior do agente. Um ambiente não-MDP ocorre quando o espaço de entrada é apenas parcialmente observável. Para ambientes não-MDP, podem ser definidas arquiteturas que estimem os estados não-observados com base nos estados e nas ações passadas do agente.

Uma análise ampla dos modelos disponíveis para realização de treinamento condicionado, como a apresentada em [32], nos mostra uma base comum a todos os algoritmos tradicionais:

- um mecanismo de exploração, no qual a ação A é definida com objetivo de explorar as funções R e T ;
- um mecanismo de escolha, no qual a ação A é definida em função do ganho R e do estado S atual.

Em princípio, a função T não precisa ser conhecida pelo agente, mas se torna importante para o caso de ambientes que gerem reforços com atraso, pois permite a propagação de um reforço recebido em um dado tempo para os estados anteriores.

Os vários modelos tradicionais de treinamento condicionado existentes diferem, basicamente, quanto aos seguintes aspectos:

- estratégias de otimização do uso dos mecanismos de exploração;
- formas de armazenamento e cálculo estatístico da função R ;
- formas de utilização da função T ;
- formas de propagação de reforços com atraso;

- formas de implementação (simbólica ou conexionista);
- estratégias de cálculo estatístico que permitam a evolução das funções R e T , quando estas variam com o tempo.

De modo geral, todos estes modelos levam a uma função que pode ser representada por:

$$a = F(s) \quad (4.1)$$

onde F é uma função que leva a uma saída a , para uma dada entrada s , comandada pela função de reforço R .

Para espaço com elevadas dimensões, pode-se subdividir F em um conjunto de funções que mapeiem subespaços de S para saídas intermediárias, que são combinadas para gerar o espaço de A , como mostrado pela Fig. 6.

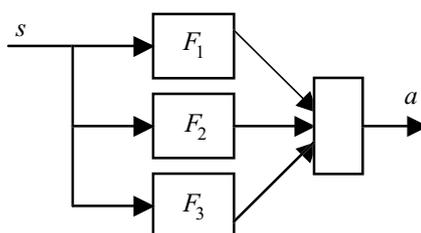


Fig. 6. Modelo de controle hierárquico.

4.5 Modelo Proposto

Tanto as ferramentas para modelagem de sistemas quanto os sistemas de controle tradicionais são basicamente modelados por uma função que considera os seguintes elementos:

Entrada → *Processamento* → *Saída*

Associados às entradas, podemos ter também sinais de referência que definem estados desejados para essas entradas. Nesse modelo, existe uma forte e clara distinção entre os espaços de entrada e saída e sempre podemos definir uma função que mapeie um

espaço em relação ao outro.

O modelo proposto quebra este paradigma de controle, pois nele os espaços de entrada e saída são tratados de forma indistinta, sendo substituídos pelo conceito de “espaço de causa” e “espaço de efeito”, em que cada um deles pode conter tanto sinais de entrada quanto de saída.

Além disso, o novo modelo processa uma nova classe de sinais que podem ser definidos como “objetivos de entrada” e “objetivos de saída”. Esses objetivos não são sinais convencionais, pois, enquanto um sinal de entrada em um dado tempo irá representar um ponto no espaço, um sinal objetivo pode representar um conjunto de pontos ou até mesmo uma região de um espaço. Dessa forma, mesmo operações básicas, como a soma ou a negação de objetivos, são realizadas de maneira totalmente diversa das operações realizadas sobre sinais convencionais de entrada e saída. Os objetivos podem ser propagados através de estruturas hierárquicas muito mais poderosas do que as estruturas de controle tradicionais.

O novo modelo está baseado em uma estrutura de processamento que foi denominada “esquema de transição”. Essa nova estrutura foi definida a partir do modelo de “esquemas mentais” originalmente proposto por Piaget. Esse modelo é descrito a seguir com maiores detalhes.

4.6 Modelos de Esquemas

O conceito de esquemas foi introduzido por Piaget na década de 50, através de uma proposta para modelagem da capacidade cognitiva de um ser humano, dada por uma coleção de “esquemas mentais” ou “esquemas de assimilação”, assim definidos:

- objetos e situações aos quais o esquema se aplica;
- ações a serem tomadas;
- objetivos ou resultados previstos;
- parâmetros de avaliação.

A fim de esclarecer um pouco mais a proposta de “esquemas mentais”, definidos por Piaget, consideremos o seguinte exemplo:

Esquema: “abrir uma garrafa”

- objetos e situações aos quais o esquema se aplica: “quando desejar tirar a tampa de uma garrafa”;
- ações a serem tomadas: “pegar um abridor de garrafas e usá-lo sobre a tampa”;
- objetivos ou resultados previstos: “garrafa destampada”;
- parâmetros de avaliação : “verificar se a tampa saiu”.

Quando este esquema for aplicado a uma garrafa de vinho não irá funcionar, pois, para esse caso, precisaremos de um saca rolhas. Dessa forma, o esquema “abrir uma garrafa” poderá ser especializado (dividido) em dois novos esquemas: “abrir uma garrafa fechada com uma tampinha metálica” e “abrir uma garrafa fechada com rolha” que terão, como objetos, ações distintas (por exemplo: usar o abridor ou o saca-rolhas) que levarão a um mesmo objetivo.

Dessa forma, segundo Piaget, os seres humanos deveriam possuir um conjunto muito grande de “esquemas mentais” que seriam constantemente reavaliados, especializados e assimilados (divididos e agrupados) a fim de responder a todas as nossas possíveis ações perante as situações com que nos defrontamos no dia-a-dia.

A proposta de Piaget não foi formulada em termos matemáticos precisos e, desde então, alguns autores (Balkenius [40] e Sun [41], dentre outros) propuseram modelos de esquemas matematicamente mais elaborados, porém muitas vezes não tão abrangentes. Recentemente, Drescher [42] e Wazlawick [43] introduziram novos modelos, buscando formalizar a idéia original de esquemas de assimilação, apresentada por Piaget, principalmente no que se refere à divisão e ao agrupamentos de esquemas em estruturas que foram denominadas Metaesquemas. Dois (ou mais) esquemas podem ser associados (em série ou paralelo) a fim de criar metaesquemas, os quais podem, por sua vez, serem novamente associados, criando uma hierarquia de metaesquemas.

Como a proposta original de Piaget define um modelo genérico (mas incompleto) para a organização da informação no cérebro humano, ainda teremos um grande caminho a percorrer até que surja um modelo que seja efetivamente completo, já que tal modelo deve permitir a emulação de algumas das características mais importantes de um cérebro humano.

É interessante notar que existe uma semelhança entre os esquemas propostos por Piaget e o modelo MDP de Markov. Os dois modelos são diferentes, pois o modelo de

Markov é voltado para a descrição de um ambiente e o de Piaget procura descrever como um ser humano organiza a informação sobre o ambiente que o cerca.

Desta forma, a função $T(s, a, s')$ é válida para todos os estados de entrada e saída e tem um grau de probabilidade associado à cada transição. Já o conjunto de esquemas representa apenas um modelo interno, portanto imperfeito, que o agente criou do ambiente. Além disso, os esquemas de Piaget podem ser modificados, especializados, generalizados, criados e destruídos, o que não acontece com o MDP.

4.7 Esquemas de Transição

O presente trabalho está baseado em uma evolução do conceito de “esquemas de assimilação” definidos por Piaget [24] e também no modelo MDP de Markov [39], que denominamos “esquemas de transição” [44].

Definição 5.1 - Dados dois subespaços quaisquer, compostos por sinais de entrada e/ou de saída, um deles é especificado como espaço de causa $C = \{c_1, c_2, \dots, c_n\}$ e o outro, como espaço de efeito $E = \{e_1, e_2, \dots, e_m\}$. Um esquema de transição é definido pela seguinte função:

$$ET(c_i, c_f, e_i, e_f, t, p, r) \quad (4.2)$$

onde:

- c_i é o estado inicial do espaço de causa;
- c_f é o estado final do espaço de causa;
- e_i é o estado inicial do espaço de efeito;
- e_f é o estado final do espaço de efeito;
- t é o tempo médio de transição;
- p é a possibilidade de transição²;
- r é o parâmetro de avaliação do esquema.

² Como o agente cognitivo tem uma percepção incompleta do sistema no qual está inserido, não é recomendável utilizar o conceito de probabilidade (de transição), mas sim o conceito de possibilidade definido por Zadeh em [58], que se adapta bem melhor ao modelo aqui proposto.

Um esquema de transição é criado quando uma mudança de estado ocorre em C e, a seguir (após um tempo t), uma mudança de estado é observada em E . O parâmetro p é calculado através da observação das transições em C que levam a uma mesma transição em E .

O parâmetro de avaliação r é opcional, sendo porém importante quando o ambiente varia de forma abrupta e alguns esquemas deixam de funcionar (o valor da possibilidade de transição irá decrescer muito lentamente quando um esquema treinado deixar de funcionar). Dessa forma, toda vez que um esquema é atualizado seus resultados podem ser reavaliados, assim o valor de r irá evoluir com o tempo, podendo indicar que um esquema não é mais válido e deve ser destruído ou adaptado. Esse parâmetro é utilizado na escolha de esquemas específicos dentro de um conjunto de esquemas disponíveis.

A Fig. 7 apresenta uma ilustração deste conceito para um espaço bidimensional, onde os estados são representados por pontos e as transições por flechas. Nesse caso, uma transição no espaço de causa gera uma transição no espaço de efeito.

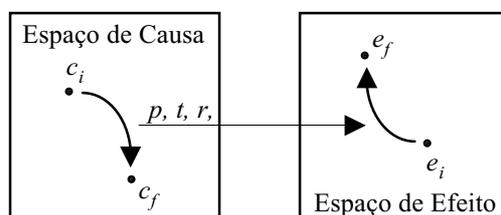


Fig. 7. Representação de um esquema de transição.

4.8 Classificação dos Esquemas de Transição

Os esquemas de transição podem ser classificados em função do tipo de transição identificada. Como no estudo das transições, podemos identificar duas condições para cada espaço, ou seja: ocorreu e não ocorreu uma transição, poderíamos, em princípio, ter quatro tipos de esquemas de transição. No entanto, o estado em que nenhuma transição ocorreu, nos dois espaços, não é considerado um esquema de transição. Assim, ficamos com apenas três tipos de esquemas de transição:

- esquema de transição completo ($c_i \neq c_f$ e $e_i \neq e_f$), em que ocorrem transições nos dois espaços (mesmo que não sejam simultâneos);

- esquema de transição sem efeito observável ($c_i \neq c_f$ e $e_i = e_f$), em que, para um tempo t maior do que um limite máximo, nada ocorre no espaço de efeito;
- esquema de transição sem causa observável ($c_i = c_f$ e $e_i \neq e_f$), em que um efeito é percebido sem que nada varie no espaço de causa.

Nos esquemas sem efeito observados, uma variação no espaço de causa aparentemente não traz nenhuma consequência observável para o estado de causa atual. Nos esquemas sem causa observável, as transições no espaço de efeito podem estar ocorrendo de forma aleatória ou, como o próprio nome já diz, são devidas a um efeito não observável.

É importante notar que todos os três tipos de esquemas podem estar presentes durante o tempo em que os dois espaços estiverem sendo observados. Se não existirem ligações efetivas de causa e efeito entre os subespaços C e E , poucos esquemas de transição completos serão observados.

Assim, o tipo de esquema mais importante para um modelo de controle é o esquema de transição completo. Normalmente, uma variação no espaço de causa, seguida por uma variação no espaço de efeito, indica que possivelmente a transição de efeito se deve à variação identificada no espaço de causa.

Considerando apenas os esquemas de transição completos, podemos verificar, na prática, que uma mesma transição de efeito pode ser gerada por várias transições de causa distintas. Por exemplo, no controle da mão de um robô, em que o espaço de efeito seja composto pelo estado da mão (aberta ou fechada) e pela posição da mão (um conjunto de locais específicos), e seja dado por um sensor de tato (mão segurando um objeto ou mão vazia), a transição de efeito de “mão vazia” para “mão segurando o objeto” poderá ser obtida quando ocorrer uma transição de causa “mão aberta na posição Y” para “mão fechada na posição X”. Porém, poderá ocorrer também uma transição de causa “mão aberta na posição Z” para “mão fechada na posição X”. Isso significa que o objeto está na posição X e que a posição inicial do braço (Y ou Z) não importa. Assim, se a mão estiver na posição H, não será necessário ir primeiro para a posição Y (ou para a Z), podendo-se ir direto para a posição X. Nesse caso, a transição é dita de causa convergente, pois o que importa é apenas o ponto final da transição. Assim, se analisarmos as transições dos esquemas completos, elas poderão ser também classificadas em três grupos:

- transições de causa divergente: apenas o ponto inicial da transição no espaço de causa é importante;
- transições de causa convergente: apenas o ponto final da transição no espaço de causa é importante;
- transições completas: os dois pontos do espaço de causa são importantes.

Esta classificação deve ser levada em conta pelos procedimentos de controle dos esquemas que devem definir como objetivos somente os estados de causa que forem relevantes, como, no exemplo da posição da mão, no qual o que importa é apenas a posição final. Já, no caso do estado da mão, é importante que ela esteja primeiro aberta para depois ser fechada. Assim, os dois pontos são importantes e esta é uma transição completa.

4.9 Metaesquemas de Transição

O conjunto de todos os esquemas de transição, atribuídos a dois subespaços (C , E), define um metaesquema de transição (MET). Além das entradas correspondentes aos sinais desses subespaços, um metaesquema de transição possui também uma entrada e uma saída assim definidas:

- E_e é a entrada objetivo para o espaço de efeito (de mesma dimensão que E);
- S_c é a saída objetivo para o espaço de causa (de mesma dimensão que C).

Apesar dos “sinais objetivos” terem uma dimensão fixa, eles podem, para um determinado instante, representar um conjunto de pontos ou mesmo uma região do subespaço de causa ou de efeito. De fato, os “sinais objetivos” não podem ser tratados como sinais convencionais (tradicionais) de entrada e saída. Por exemplo, a negação de um objetivo gera um conjunto de pontos igual a todo o espaço, com exceção do ponto negado. Da mesma forma, ao interligarmos (somarmos) dois sinais objetivos, obtemos um conjunto de pontos igual à soma dos conjuntos representados por cada objetivo. O metaesquema, ao receber uma entrada objetivo (E_e) válida, utiliza as informações de todos os esquemas de transição por ele observados para gerar uma saída objetivo (S_c). A Fig. 8 ilustra as entradas e saídas de um metaesquema de transição.

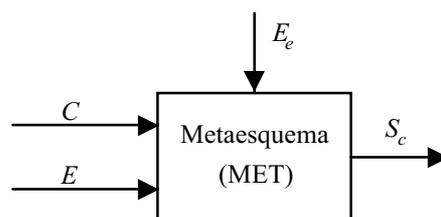


Fig. 8. Entradas e saídas de um metaesquema de transição.

A Fig. 9 ilustra uma forma de ligação a partir da qual podemos demonstrar que o modelo de controle tradicional é um caso particular do modelo de controle proposto, onde:

- E é igual ao espaço de entrada;
- C é igual a S_c , que é igual ao espaço de saída;
- E_e é igual a uma entrada de referência.

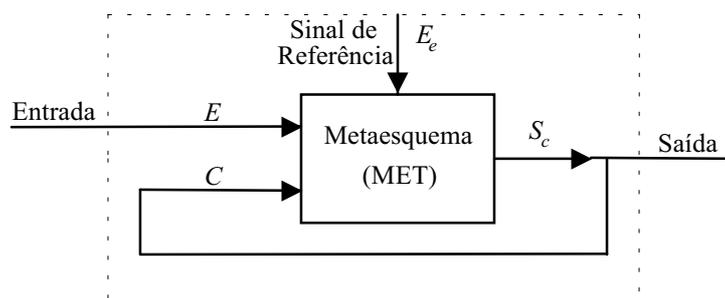


Fig. 9. Modelo de controle tradicional obtido a partir de um metaesquema de transição.

Uma das mais importantes características dos metaesquemas de transição é a sua possibilidade de interligação dos mesmos de forma hierárquica, o que permite a definição de sistemas de controle relativamente complexos a partir de blocos mais simples, como ilustrado pela Fig. 10. Nesse diagrama, os objetivos de saída de um metaesquema são conectados aos objetivos de entrada de um ou mais metaesquemas dos níveis hierárquicos que lhe são inferiores.

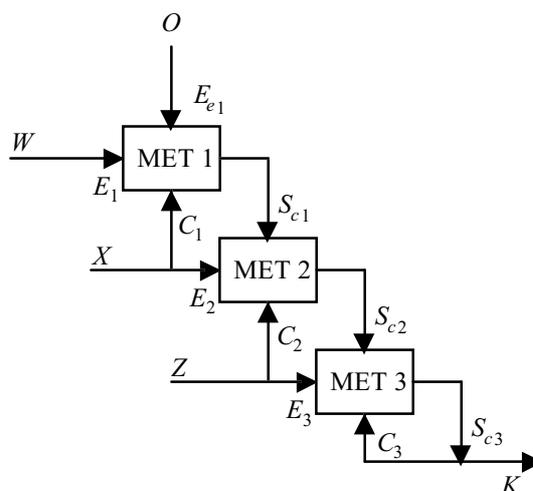


Fig. 10. Exemplo de hierarquia de metaesquemas.

É importante observar que o espaço, que é de causa para um metaesquema, torna-se de efeito para o metaesquema situado logo abaixo na hierarquia. Isso possibilita um encadeamento dos vários espaços de causa e efeito, que permite o desdobramento de objetivos mais elevados em objetivos mais detalhados e mais específicos.

4.10 Controle de um Metaesquema de Transição

Um metaesquema de transição é controlado por uma série de procedimentos:

- procedimento de manutenção dos esquemas;
- procedimento para definição de saídas objetivo;
- mecanismos de exploração.

O procedimento de manutenção de esquemas é responsável pela criação e armazenamento de todos os esquemas de transição presentes no metaesquema.

Inicialmente, o metaesquema não possui nenhum esquema de transição e os esquemas vão sendo gerados e atualizados através da observação das mudanças de estado que ocorrem nos espaços C e E . Esse procedimento é relativamente trivial, pois se deve simplesmente observar as transições e atualizar os valores de tempo médio e de possibilidade de cada uma delas.

O procedimento de definição de saídas objetivos utiliza o sinal E_e para gerar um sinal S_c , definido a partir dos esquemas de transição armazenados. Os objetivos são

propagados por uma função gradiente dada pelas possibilidades de transição no espaço E . Esse procedimento pode se tornar bastante complexo, principalmente, para poder lidar com objetivos múltiplos e com a negação de objetivos.

Os mecanismos de exploração geram transições nos diversos espaços de causa, que permitem a criação dos esquemas de transição. Essa exploração pode ser realizada de duas formas distintas:

- utilização de um mecanismo de exploração interno ao agente, que ative suas saídas de forma aleatória ou sistemática (reflete um processo de aprendizagem por tentativa e erro);
- utilização de um mecanismo de controle das saídas externo ao agente (treinamento supervisionado).

A seguir apresentaremos com maiores detalhes alguns exemplos de formas de representação e mecanismos de controle de metaesquemas de transição.

4.11 Representação de Metaesquemas de Transição

Um metaesquemas de transição (MET) é caracterizado por quatro conjuntos de sinais:

- espaço de causa (C);
- espaço de efeito (E);
- entrada de objetivos (E_e);
- saída de objetivos (S_c).

Podemos considerar que o MET lida com dois conjuntos de sinais distintos, um espaço de sinais (sinais de entrada e saída) e um espaço de objetivos (objetivos de entrada e saída, interno ao sistema), como mostra a Fig. 11.

Desta forma, o MET fica complementemente definido quando indicamos que sinais estão a ele conectados, o que pode ser feito de algumas maneiras:

- programação gráfica: através de um editor gráfico, com recursos para definição de diagramas de blocos, podemos programar a interligação de uma série de MET de forma similar ao diagrama mostrado na Fig. 11, sendo os sinais de entrada e saída, para cada bloco, facilmente configurados;

- programação textual: utilizando uma sintaxe³ de programação bastante simples, podemos definir as entradas e saídas de cada MET e, dessa forma, definir também suas interligações.

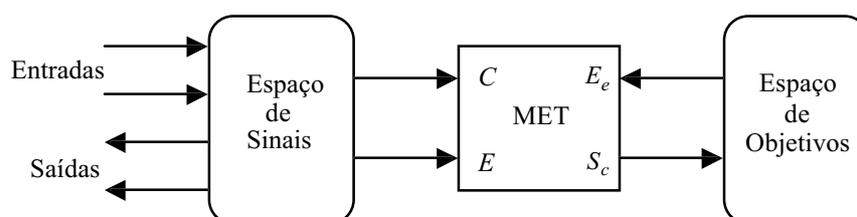


Fig. 11. Interligação de um Metaesquema de Transição.

Apesar da programação gráfica ser bastante interessante, a criação de um editor demandaria um esforço considerável sem trazer grandes inovações. A sintaxe, por sua vez, é de fácil implementação e apresenta as informações em um formato bastante compacto.

Desta forma, definimos a seguinte sintaxe:

- cada metaesquema é representado em uma única linha, através de quatro conjuntos de parâmetros: (*espaço de causa | espaço observado | saídas objetivos | entradas objetivos*);
- cada parâmetro é representado por um conjunto de dados separados por vírgulas. Cada dado indica um sinal de entrada (E) e de saída (S) ou de objetivo (O), sendo que os sinais recebem uma numeração seqüencial;
- comentários podem ser incluídos após o sinal %;
- o símbolo \$CC (causa convergente) indica que a transição inicial de causa não importa e que somente a transição final de causa deve ser armazenada;
- o comando FIM indica o final da lista de esquemas, sendo as próximas linhas ignoradas.

Para exemplificar esta sintaxe de programação, mostramos, na Tabela 4, a configuração de um conjunto de 10 metaesquemas utilizados para o aprendizado do “Jogo da Velha”. Uma descrição detalhada desse programa é apresentada no Apêndice 1.

³ Preferimos não usar o termo “linguagem de programação”, pois, apesar de ser usada para programar metaesquemas que irão aprender tarefas complexas, esta sintaxe não contém comandos nem outros elementos comuns às linguagens tradicionais.

TABELA 4
EXEMPLO DE PROGRAMAÇÃO DE UM CONJUNTO DE METAESQUEMAS DE TRANSIÇÃO

```

% Programa do "Jogo da Velha"
% Espaço de causa = E1, E2, E3, E4, E5, E6, E7, E8,E9 = tabuleiro
% Espaço de efeito = E11 E12 = dor, prazer
% Entrada de objetivos = 0, 1
% Entrada de objetivos negados = 1, 0
% Saída de objetivos = O1, O2, O3, O4, O5, O6, O7, O8, O9
% O símbolo $CC indica que a transição inicial de causa não importa
($CC, E1, E2, E3, E4, E5, E6, E7, E8,E9 | E11 E12 | O1, O2, O3, O4, O5, O6, O7, O8, O9 | 0 1 | 1 0)
% Nove MET's de controle de saída (x = 1,2,...,9)
% Espaço de causa = Sx, E10, E14 = saída x, jogar, novo jogo
% Espaço de efeito = Ex
% Entrada de objetivos = Ox
% Saída de objetivos = Sx
(S1, E10, E14 | E1 | S1 | O1)
(S2, E10, E14 | E2 | S2 | O2)
(S3, E10, E14 | E3 | S3 | O3)
(S4, E10, E14 | E4 | S4 | O4)
(S5, E10, E14 | E5 | S5 | O5)
(S6, E10, E14 | E6 | S6 | O6)
(S7, E10, E14 | E7 | S7 | O7)
(S8, E10, E14 | E8 | S8 | O8)
(S9, E10, E14 | E9 | S9 | O9)
FIM

```

4.12 Implementação de Metaesquemas de Transição

Os metaesquemas de transição podem ser implementados através de uma máquina de inferência, gerenciando uma base de dados única para todos os METs, controlada por rotinas executadas seqüencialmente. É possível também implementar uma estrutura de processamento distribuído em que cada MET representa um processo distinto dos demais, com sua própria base de dados. Dessa forma, todos os METs podem ser processados em

paralelo, sendo que, dentro de um MET, as próprias rotinas de manutenção de esquemas de definição de saídas também podem ser executadas simultaneamente. Como o modelo já considera os tempos entre transições de causa e de efeito, não é necessário que haja nenhum tipo de sincronização temporal entre os processos.

Considerando que a arquitetura proposta é preponderantemente distribuída, os esquemas de transição podem, em princípio, ser implementados através de redes neurais. Dessa forma, seriam necessários aprimoramentos nas estruturas neurais atualmente disponíveis para a efetivação dessa implementação.

Os metaesquemas utilizados neste trabalho foram definidos como uma série de processos independentes, executados em uma máquina seqüencial, sendo que as atualizações das entradas e saídas bem como o incremento do tempo são realizados em uma etapa única, de forma a simular um processamento paralelo.

4.13 Exemplo de Implementação de Metaesquemas de Transição

Os modelos e definições em que se baseiam os metaesquemas de transição contrariam um dos conceitos mais arraigados da área de análise e controle de sistemas, que é o conceito de “sinais de entrada” e “sinais de saída”. No modelo proposto, passamos a lidar com espaços de causa e efeito que são compostos de conjuntos de sinais de entrada ou de saída. Assim, podemos ter, em um mesmo espaço, sinais de entrada e de saída “misturados”, pois os METs não lidam com estes conceitos, mas sim com relações de causa e efeito, que não são necessariamente relações de causa e efeito reais em sentido físico, mas sim, relações virtuais.

Dessa forma, antes de tratar do processamento de metaesquemas, iremos recorrer a uma série de exemplos simples que facilitem a compreensão do novo modelo.

4.14 Exemplos Didáticos de Esquemas de Transição

De forma simplificada, podemos considerar que um metaesquema observa dois espaços: o espaço de causa e o espaço de efeito.

Consideremos um sistema do nosso dia-a-dia, composto por um agente cognitivo que deseja tomar banho e para tal deve controlar a temperatura de um chuveiro elétrico.

Consideremos que o agente observa a vazão e uma determinada temperatura da água e controla, com suas mãos, a abertura e o fechamento de uma torneira.

Neste contexto, podemos definir uma estrutura de metaesquemas que irão controlar este agente:

Metaesquema 1: Espaço de causa - Vazão da água

Espaço de efeito - Temperatura da água

Metaesquema 2: Espaço de causa - Grau de abertura da torneira

Espaço de efeito - Vazão da água

Metaesquema 3: Espaço de causa - Direção de giro da mão do agente

Espaço de efeito - Grau de abertura da torneira

Com o passar do tempo, os metaesquemas poderiam identificar relações do tipo:

Metaesquema 1: Aumentando a vazão, a temperatura diminui.

Diminuindo a vazão, a temperatura aumenta.

Abaixo de uma dada vazão, a temperatura fica em um valor mínimo.

Metaesquema 2: Abrindo a torneira, a vazão aumenta.

Fechando a torneira, a vazão diminui.

Metaesquema 3: Girando a mão no sentido horário, a torneira se abre.

Girando a mão no sentido anti-horário, a torneira se fecha.

Com a interligação hierárquica destes metaesquemas, o agente poderá controlar a temperatura. Essa interligação será feita através de objetivos de entrada e saída, definidos para cada metaesquema.

Uma definição completa de um metaesquema de transição pode ser observada na Fig. 12, onde são considerados quatro tipos de sinais:

- entradas de observação dos sinais do espaço de causa;
- entradas de observação dos sinais do espaço de efeito;
- entradas de objetivos para o espaço de efeito;
- saída de objetivos para o espaço de causa.

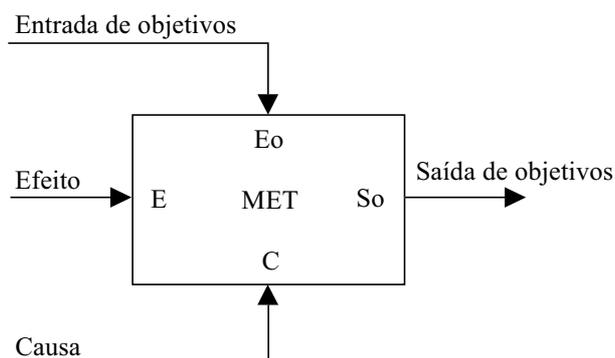


Fig. 12. Representação completa de um metaesquema.

Assim, em nosso exemplo, podemos definir os metaesquemas como segue:

Metaesquema 1: Espaço de causa - Vazão da água

Espaço de efeito - Temperatura da água

Entradas de objetivos - Temperatura objetivo

Saídas de objetivos - Vazão objetivo

Metaesquema 2: Espaço de causa - Grau de abertura da torneira

Espaço de efeito - Vazão da água

Entradas de objetivos - Vazão objetivo

Saídas de objetivos - Abertura da torneira objetivo

Metaesquema 3: Espaço de causa - Direção de giro da mão do agente

Espaço de efeito - Grau de abertura da torneira

Entradas de objetivos - Abertura da torneira objetivo

Saídas de objetivos - Giro da mão objetivo

Desta forma, ao definir cada metaesquema, já definimos sua interligação hierárquica, como mostrado na Fig. 13.

É importante também observar que definimos três metaesquemas a fim de sermos mais didáticos, pois poderíamos utilizar apenas dois, ou mesmo um metaesquema para controlar a temperatura do chuveiro. Isso pode ser feito relacionando diretamente a temperatura com a posição da torneira ou ainda a temperatura com o giro da mão.

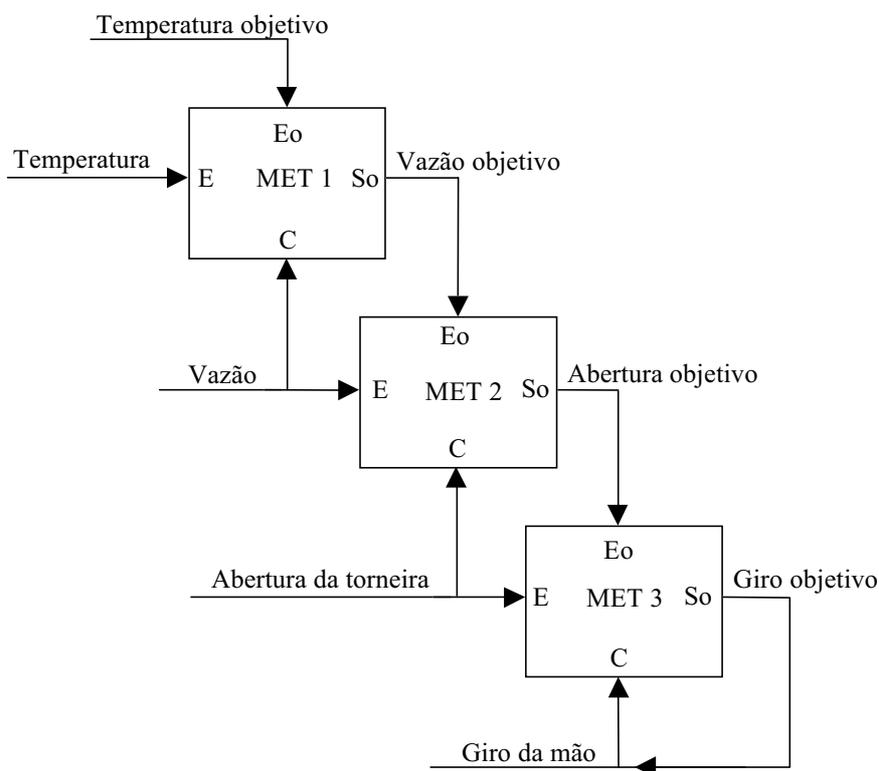


Fig. 13. Interligação dos metaesquemas para controle de temperatura.

O sistema de controle assim definido irá operar da seguinte forma:

- uma temperatura objetivo é definida de alguma forma (por exemplo, por meio de sinais de reforço do “corpo” do agente que julgue a água muito quente ou muito fria);
- o MET1 recebe esta temperatura objetivo e gera uma vazão objetivo;
- o MET2 recebe esta vazão objetivo e gera uma abertura da torneira objetivo;
- o MET3 recebe abertura da torneira objetivo e gera um giro da mão objetivo;
- como o giro da mão é um sinal de saída, um “giro da mão objetivo” é atingido, definindo-se o valor de saída como sendo igual ao valor objetivo.

Assim, o agente irá girar a mão para variar a temperatura de forma a atingir o valor de temperatura objetivo.

Antes de considerar as formas de processamento dos METs, é necessário aprofundar um pouco mais as considerações sobre a definição de espaços de causa e de efeito. A fim de mostrar a flexibilidade dessas definições, vamos construir uma nova estrutura de metaesquemas (considerando um sistema externo ao agente), na qual

invertemos os espaços de causa e efeito:

Metaesquema 1: Espaço de causa - Temperatura da água

Espaço de efeito - Vazão da água

Metaesquema 2: Espaço de causa - Vazão da água

Espaço de efeito - Grau de abertura da torneira

Metaesquema 3: Espaço de causa - Grau de abertura da torneira

Espaço de efeito - Direção de giro da mão do agente

Estes novos metaesquemas com o passar do tempo irão identificar as seguintes relações:

Metaesquema 1: Aumentando a temperatura, a vazão diminui.

Diminuindo a temperatura, a vazão aumenta.

Metaesquema 2: Aumentando a vazão, a torneira se abre.

Diminuindo a vazão, a torneira se fecha.

Metaesquema 3: Abrindo a torneira, a mão do agente gira no sentido horário.

Fechando a torneira, a mão do agente gira no sentido anti-horário.

Estas relações observadas são tão válidas como as dos primeiros metaesquemas apresentados, mas o novo conjunto de metaesquemas é ligado de uma forma hierárquica diferente e, dessa forma, temos duas configurações distintas:

- no primeiro caso, o agente irá girar a mão para controlar a temperatura do chuveiro;
- no segundo caso, um sistema externo (por exemplo, um outro agente) irá variar a temperatura visando controlar o giro da mão do agente.

A segunda situação não é totalmente absurda, pois, se a temperatura do chuveiro puder ser controlada por um sistema externo (variando por exemplo a tensão na resistência de aquecimento), esse sistema poderia de fato “controlar” a mão do agente, enquanto ele tenta obter uma temperatura que lhe seja adequada. Esse exemplo, embora bastante acadêmico, mostra que as relações de causa e efeito impostas aos metaesquemas não precisam ser necessariamente relações reais (em sentido físico) de causa e efeito. Na realidade, não é necessário que sempre existam relações entre os sinais de causa e os de efeito. O exemplo a seguir mostra uma condição em que uma “falsa” causa foi considerada

na montagem do metaesquema:

Metaesquema 1: Espaço de causa - Grau de abertura da torneira

Estado da luz do banheiro

Espaço de efeito - Temperatura da água

Metaesquema 2: Espaço de causa - Direção de giro da mão

Espaço de efeito - Grau de abertura da torneira

Ora, em princípio, o fato da luz do banheiro estar acesa ou apagada não afeta a temperatura do chuveiro e, nesse caso, o metaesquema logo identificará uma correlação muito baixa entre essa variável e a temperatura. Dessa forma, as ligações “não consistentes” de causa e efeito são naturalmente eliminadas. Logicamente, se nenhuma relação for observada, o metaesquema se tornará inócuo e inoperante.

Um outro ponto relacionado com a definição dos espaços de causa e efeito é o fato de que, muitas vezes, não temos acesso a todos os sinais de um dado sistema e, dessa forma, os espaços de causa e efeito podem conter estados não observados. De forma geral, causas não observadas irão gerar eventos “aleatórios” para o sistema e efeitos não observados irão limitar as estratégias de controle, podendo comprometer a operação do sistema. Em uma estrutura hierárquica de metaesquemas, em que os espaços são sucessivamente causas e efeitos, uma variável não observada poderá gerar simultaneamente uma causa e um efeito não observados.

No exemplo do controle do chuveiro, temos algumas variáveis não observadas como, por exemplo, o “estado da resistência elétrica do chuveiro” (normal ou “queimada”) e o “estado do fornecimento de energia” (normal ou sem energia). Esses dois sinais serão causas não observadas que poderão gerar um “problema” para o metaesquema, pois, em uma condição de defeito, a temperatura da água fica fixa em um valor mínimo. Nessa condição, podemos considerar também que temos um efeito não observado que seria representado pelo “estado do chuveiro” (normal ou com defeito) que também não é lido diretamente pelo sistema.

Um sistema efetivo de controle de metaesquemas deve ser capaz de lidar com causas e efeitos não observados. Isso pode ser feito através da criação de variáveis internas a fim de estimar sinais não observados. Essa criação pode ser feita basicamente de duas maneiras:

- verificação de condições anormais criando automaticamente condições de falhas não observadas e definindo seu estado como “falha detectada”;
- utilização de treinamento supervisionado para informar ao sistema o estado de variáveis não observadas.

Para uma melhor compreensão do exposto acima e visando expandir a análise de metaesquemas de transição, iremos elaborar um exemplo um pouco mais completo.

Consideremos que o agente deva, além do chuveiro, controlar também uma lâmpada, podendo existir algumas possibilidades de falha:

- queima da resistência do chuveiro;
- desligamento do disjuntor do chuveiro por sobrecarga;
- queima da lâmpada;
- falta de energia.

Consideremos ainda que o agente possa realizar certas ações:

- controlar a torneira do chuveiro;
- acionar o interruptor da lâmpada;
- religar o disjuntor do chuveiro;
- trocar a lâmpada;
- trocar a resistência do chuveiro.

Para controle deste agente, podemos definir os seguintes metaesquemas de transição:

Metaesquema 1: Espaço de causa - Grau de abertura da torneira

Estado da variável “problemas no chuveiro”

Espaço de efeito - Temperatura da água

Metaesquema 2: Espaço de causa - Direção de giro da mão do agente

Espaço de efeito - Grau de abertura da torneira

Metaesquema 3: Espaço de causa - Troca da resistência do chuveiro

Espaço de efeito - Estado da resistência do chuveiro

Metaesquema 4: Espaço de causa - Religar o disjuntor do chuveiro

Espaço de efeito - Estado do disjuntor do chuveiro

- Metaesquema 5: Espaço de causa - Estado da resistência do chuveiro
Estado da variável “problemas na lâmpada”
Estado do disjuntor do chuveiro
Espaço de efeito - Estado da variável “problemas no chuveiro”
- Metaesquema 6: Espaço de causa - Estado do interruptor da lâmpada
Estado da variável “problemas na lâmpada”
Espaço de efeito - Estado da lâmpada.
- Metaesquema 7: Espaço de causa - Ação da mão do agente sobre o interruptor
Espaço de efeito - Estado do interruptor da lâmpada
- Metaesquema 8: Espaço de causa - Troca da lâmpada
Espaço de efeito - Estado de funcionamento da lâmpada
- Metaesquema 9: Espaço de causa - Estado de funcionamento da lâmpada
Estado da variável “problemas no chuveiro”
Espaço de efeito - Estado da variável “problemas na lâmpada”

Neste exemplo, os metaesquemas 1 e 2 são usados para operar o chuveiro. A variável “problemas no chuveiro” é uma causa não observada que será estimada pelo metaesquema 1 com uma regra do tipo “com a torneira aberta uma baixa temperatura indica um problema no chuveiro”. Os metaesquemas de 3, 4 e 5 indicam as ações a serem tomadas para manter o chuveiro funcionando. De maneira análoga, os metaesquemas 6, 7, 8 e 9 controlam a lâmpada e a sua manutenção.

A seguir veremos com maiores detalhes os mecanismos de treinamento e de controle dos metaesquemas de transição. Mesmo sem ter feito essa análise, a título ilustrativo, iremos antecipar o comportamento desse agente como segue.

O agente controlado pelos metaesquemas acima definidos terá uma série de objetivos, dentre eles o de controlar a lâmpada e o chuveiro e também o de mantê-los sem problemas através de ações corretivas. Uma característica importante das estruturas de controle baseadas em metaesquemas é a definição de hierarquias que podem processar múltiplos objetivos. Dessa forma, o objetivo de controlar a temperatura será automaticamente bloqueado quando um problema no chuveiro for identificado. Da mesma forma, o objetivo de manter o chuveiro funcionando está sempre satisfeito até o momento em que uma falha seja identificada.

Após um período de treinamento, o agente irá descobrir que, quando a lâmpada e o chuveiro apresentam problemas simultaneamente, as ações corretivas não são eficientes (pois ocorreu uma falta de energia). Dessa forma, ao detectar um problema no chuveiro, o agente tentará ligar a lâmpada (e *vice-versa*), a fim de saber se o problema pode ser corrigido ou se apenas trata-se de uma falta de energia. Além disso, o agente poderá verificar outros pontos como, por exemplo: a ocorrência de problemas no disjuntor é maior do que a na resistência do chuveiro. Assim, ele irá primeiro religar o disjuntor e, se o chuveiro continuar não funcionando, ele irá trocar a resistência.

Para completar esta visão geral da forma como operam os metaesquemas no controle de um agente cognitivo, podemos imaginar um agente que, além dos metaesquemas citados, possua um conjunto muito maior de esquemas, que formariam uma vasta hierarquia para a qual o objetivo maior do agente fosse, por exemplo, “viver bem”. Esse objetivo seria transferido por meio de metaesquemas para outros objetivos, tais como: “alimentar-se”, “ganhar dinheiro”, “tomar banho”, “tomar água”. Um objetivo específico, como “tomar banho”, poderia gerar cadeias de objetivos, tais como: “ir para o banheiro”, “tirar as roupas”, “controlar a temperatura da água”. O objetivo de “controlar a temperatura” poderia ser substituído pelo “consertar o chuveiro”, em caso de uma falha no chuveiro. O objetivo “consertar chuveiro” poderia disparar o objetivo “comprar uma nova resistência”, que poderia gerar novos objetivos como “dirigir até o mercado” ou “pegar dinheiro no caixa eletrônico”; cada um dos quais deve ser “quebrado” em objetivos menores até definir ações individuais. Essa hierarquização de objetivos nos parece bastante natural, permitindo que tarefas muito complexas sejam divididas em tarefas menores e que objetivos genéricos gerem objetivos específicos. É importante salientar, em princípio, que todos os metaesquemas estão ativos simultaneamente, sendo que cada um regula a sua própria ativação sem nenhuma necessidade de um sistema de controle centralizado e nem de sinais de sincronismo temporal. Além disso, os metaesquemas seriam constantemente criados, adaptados, subdivididos e agrupados ao longo da vida do agente, passando por um processo de treinamento contínuo.

4.15 Definição de Metaesquemas de Transição

Um aspecto importante dos sistemas de controle baseados em metaesquemas de transição é que toda a “programação” necessária se limita à definição dos sinais que estão conectados a cada esquema e, por consequência, à definição da sua hierarquia, como foi mostrado na Fig. 13. Podemos observar que a quantidade de informação necessária para essa “programação” é bastante reduzida, podendo ser comparada a uma simples definição de cadeias de objetivos. No presente trabalho, os metaesquemas utilizados foram definidos manualmente, mas nada os impede de utilizar técnicas para a definição automática dessas estruturas como, por exemplo, a aplicação de algoritmos genéticos para evolução de esquemas gerados aleatoriamente.

De um ponto de vista de plausibilidade biológica, se imaginarmos estruturas de metaesquemas criadas a partir de RNAs, veremos que a definição dos metaesquemas é dada pela própria interligação das redes entre si e com os sinais sensoriais de entrada e saída. Dessa forma, um agente cognitivo baseado em metaesquemas neurais seria inicializado com essas redes já interligadas. Muitas ações instintivas, reflexas e de controle autônomo podem ser facilmente modeladas com metaesquemas de transição. Um campo de pesquisa muito interessante seria o da utilização de algoritmos genéticos para criação de estruturas baseadas em metaesquemas neurais. Em princípio, poderíamos partir de sistemas muito simples que iriam evoluir para sistemas mais complexos com uma forte plausibilidade biológica.

De qualquer forma, como podemos observar pelo “programa” do “Jogo da Velha”, apresentado na Tabela 4, a programação de sistemas baseados em metaesquemas de transição é algo bastante diferente de todas as linguagens de programação de que temos conhecimento.

4.16 Manutenção de Metaesquemas de Transição

Um metaesquema de transição, por definição, está sempre sendo submetido a um processo contínuo de treinamento. Uma rotina de criação de esquemas de transição observa os espaços de causa e efeito, registrando mudanças nesses estados. Em princípio, cada mudança faz parte de uma transição. Procuramos sempre relacionar transições no espaço de causa com transições no espaço de efeito, associando tempos médios de atraso e

também possibilidades de transição.

Para exemplificar, consideremos os seguintes espaços:

- espaço de causa composto de quatro estados: A, B, C e D;
- espaço de efeito composto também de quatro estados: X, Y, Z e W.

Observando os dois espaços, podemos obter uma tabela das transições observadas em função do tempo, como as mostradas na Tabela 5.

TABELA 5
EXEMPLO DE TRANSIÇÕES OBSERVADAS POR UM METAESQUEMA

Número	Estado Inicial	Estado Final	Tempo
1	Z	Y	0,000
2	A	C	0,010
3	Y	Z	1,023
4	Z	X	1,582
5	X	Z	2,107
6	C	B	2,127
7	Z	Y	3,520
8	Y	W	3,702
9	B	C	3,727
10	W	X	4,121
11	X	Z	5,612
12	C	A	5,617
13	Z	X	5,980
14	X	Z	6,120
15	A	D	6,130
16	D	A	6,533
17	Z	X	7,350
18	X	Z	7,580
19	Z	Y	8,120
20	A	C	8,140

Da Tabela 6, podemos selecionar uma série de transições de causa e efeito que serão computadas pelo metaesquema.

O tempo de transição é obtido pelo tempo médio entre a transição no espaço de causa e a transição no espaço de efeito. O número de ocorrências conta quantas vezes uma dada transição foi observada. A possibilidade de transição é dada pelo número de vezes que uma transição de causa gera uma mesma transição de efeito para um dado estado de causa inicial. Transições sem causa são registradas com valor de tempo de transição igual ao tempo médio de permanência no estado e sem valor de possibilidade. Transições sem efeitos observáveis são registradas sem valores de tempo e de possibilidade.

Como demonstra este exemplo, o processo de identificação das transições é relativamente simples, mas as transições sem causa ou sem efeito observável podem gerar alguns problemas quando aparecem seguidamente, podendo levar a falsas transições. Com o passar do tempo, essas falsas transições serão eliminadas por apresentarem um baixo valor de possibilidade ou ainda por terem um número muito pequeno de ocorrências.

TABELA 6

ESQUEMAS DE TRANSIÇÃO OBSERVADOS A PARTIR DAS TRANSIÇÕES INDIVIDUAIS MOSTRADAS NA TABELA 5

Espaço de Causa		Espaço de Efeito		Tempo	Ocorrências	Possibilidade
Causa Inicial	Causa Final	Efeito Inicial	Efeito Final			
Z	X	A	A	-	1	-
X	Z	A	A	-	1	-
Z	Y	A	C	0,015	2	100%
X	Z	A	D	0,013	1	50%
Z	Y	B	B	-	1	-
Y	W	B	C	0,025	1	100%
X	Z	C	A	0,005	1	50%
X	Z	C	B	0,020	1	50%
Y	Z	C	C	-	1	-
Y	Z	C	C	-	1	-
W	X	C	C	-	1	-

É interessante também estabelecer um tempo máximo aceitável acima do qual as transições no espaço de efeito são consideradas como transições sem causa observável.

Na Tabela 6, observamos três valores de possibilidade diferentes de 100%. O primeiro ocorre na transição *A-D*, a partir de uma transição de causa *X-Y*; se observarmos na Tabela 5, veremos que a transição número 18 não gerou nenhum efeito e isso indica uma “falha” aleatória devido a um estado de causa não observado, que bloqueou o efeito dessa transição. Os outros valores são encontrados nas transições *C-A* e *C-B*, que ocorreram a partir de uma mesma transição de causa devido a um estado de causa não observado que modificou o efeito dessa transição.

Como os ambientes de treinamento utilizados neste trabalho são relativamente estáticos, não utilizamos o parâmetro de avaliação do esquema proposto no modelo apresentado na Seção 4.7. Essa simplificação facilita a implementação das estruturas de controle de metaesquemas sem levar a qualquer perda de generalidade.

Outro fator que deve ser identificado é a existência de transições de causa divergente (somente a causa inicial é importante) ou de causa convergente (somente a

causa final é importante), pois elas serão tratadas de forma distinta pelos mecanismos de controle do metaesquema. Esse tipo de transição pode ser facilmente detectado a partir de uma análise das transições geradas ou pode ser informado como um parâmetro do metaesquema.

A Tabela 7 mostra o pseudocódigo de uma rotina de manutenção de esquemas bastante simples (ela não identifica o tempo de transições sem causa observável e nem transições sem efeito observável), mas serve como base para rotinas mais completas. A função “armazenar esquemas” guarda dados dos esquemas em uma estrutura de memória (semelhante à Tabela 6), verificando quais esquemas já existem e calcula as possibilidades em função dos esquemas anteriormente identificados.

TABELA 7
PSEUDOCÓDIGO PARA MANUTENÇÃO DE ESQUEMAS

```

ci = EntradaCausa
Repetir
cf = EntradaCausa
Se cf <> ci Então
ei = EntradaEfeito
ti = LerTempo
Repetir
ef = EntradaEfeito
dt = LerTempo-ti
Até (ef <> ei) ou (dt > TempoMáximo)
Se (ef <> ei) Então
ArmazenarEsquema (ci, cf, ei, ef, dt)
ci = cf
FimSe
Até FimDoPrograma

```

4.17 Mecanismos de Exploração

Uma estrutura de controle baseada em metaesquema de transição depende fundamentalmente de um processo de treinamento para poder operar. Esse treinamento consiste basicamente na exploração dos espaços de causa e efeito através de transições semelhantes às da Tabela 5. Essa exploração pode ser realizada em parte pelo agente, através da ativação de suas saídas, mas também depende de fatores ambientais que não são controlados pelo agente. Cada ação do agente pode gerar estímulos de reforço positivo e

negativo. Esses estímulos são recebidos apenas por alguns metaesquemas que os tratam como se fossem entradas normais. Os estímulos serão utilizados como objetivos gerais para o agente que, dessa forma, evitará as ações que geram estímulos negativos e realizará as que geram estímulos positivos.

Existem diversas formas de implementação de mecanismos de exploração:

- exploração aleatória;
- exploração com busca em profundidade ou busca em largura;
- exploração (treinamento) supervisionada.

Nos experimentos realizados, utilizamos alguma exploração aleatória com predominância da exploração supervisionada. Nesse tipo de exploração, um “mentor” assume momentaneamente os controles de saída do agente, realizando as tarefas que deseja ensinar ao agente. Essa abordagem equivale ao que fazemos quando, por exemplo, seguramos na mão de uma criança para ensiná-la a escrever. Esse tipo de treinamento supervisionado é ideal para problemas em que o conjunto de ações compõem um espaço muito grande para ser explorado de forma aleatória e também para os casos em que certas ações poderiam “danificar” o agente (como no exemplo do controle de um braço de robô).

É importante lembrar que os mecanismos de exploração devem ser “desligados” depois de algum tempo para que o agente possa ativar suas saídas de acordo com seus objetivos. Em ambientes dinâmicos, esses mecanismos devem ser reativados de tempos em tempos a fim de permitir a exploração de novas características do ambiente. Isso pode ser feito, principalmente, quando o agente estiver inativo devido à falta de um objetivo imediato. Em uma estrutura mais elaborada, podemos visualizar agentes cujos mecanismos de exploração são também controlados por metaesquemas. Dessa forma, mesmo quando o agente tiver atingido todos os seus objetivos, ainda restará o objetivo de “explorar o ambiente”.

É interessante observar que, nos seres humanos, algumas vezes ficamos na dúvida entre a realização de ações conhecidas e a exploração de novas ações como, por exemplo, pegar um novo caminho para ir ao trabalho ou comprar uma nova marca de xampu em vez da habitual. Um agente complexo, implementado através do modelo proposto, também teria esse tipo de comportamento.

4.18 Processamento de “Objetivos”

A grande diferença entre o modelo de controle proposto e os modelos tradicionais está no fato de não processarmos sinais, mas sim, processarmos “objetivos”. Vejamos agora algumas características que diferenciam esses dois tipos de entidades.

Um sinal pode ser modelado como uma variável representada em um espaço (de uma ou mais dimensões) que assume um determinado valor (dentro de um conjunto de valores) para cada valor da variável independente. Um “objetivo”, por sua vez, é representado por um conjunto de valores pretendidos para um determinado sinal, esse conjunto pode ser nulo ou pode conter infinitos valores. Os objetivos podem ser discretizados e, dessa forma, o espaço associado a eles deve ser dividido em um número finito de regiões. Além disso, cada objetivo pode ser “destruído” ou suspenso por quem o gerou e também pode ser “destruído” por quem o processou.

Um sinal é propagado através de uma função de transferência. Esse tipo de função recebe um “sinal de entrada” e gera um “sinal de saída” a ela associada. Mesmo que a função seja não-linear e varie com o tempo, normalmente, teremos uma forte ligação entre as variações do sinal de entrada e o de saída. Para propagar um mesmo sinal que seja gerado em estruturas distintas, devemos utilizar algum tipo de função para combinar os dois sinais (uma operação de soma ou uma função lógica).

Um “objetivo” é propagado através de um metaesquema, sendo que um “objetivo” de entrada pode, em princípio, gerar vários “objetivos” de saída ou ainda, quando se considerar que o objetivo de entrada é inacessível ou que o mesmo já foi atingido, não gerar nenhum “objetivo”. Para propagar um mesmo “objetivo” que seja gerado em estruturas distintas, simplesmente unimos os dois conjuntos de objetivos.

Um objetivo de entrada está sempre relacionado a um espaço de efeito e um objetivo de saída está sempre relacionado a um espaço de causa. No âmbito do controle de um agente, os espaços de causa e efeito podem conter sinais de entrada ou de saída ou ainda sinais internos ao agente.

Quando um “objetivo” de saída estiver relacionado a um sinal de saída, ele é atingido simplesmente definindo o valor dessa saída como sendo igual ao do objetivo. Quando um objetivo de saída estiver relacionado a um sinal de entrada ou a um sinal interno, ele deve ser propagado através de outros metaesquemas que possuam alguns

desses sinais como entrada de “objetivo”. Dessa forma, cada objetivo será propagado até gerar um valor de saída.

Assim, percebemos que o processamento de objetivos representa uma expansão do modelo de processamento de sinais, que nos leva a uma série de novos conceitos e também a novos tipos de estruturas de processamento.

4.19 Processamento Básico de Metaesquemas de Transição

A função primordial de um metaesquema é receber objetivos de entrada e gerar objetivos de saída, com base no conjunto de esquemas de transição identificados no processo de treinamento. Esse processamento pode se tornar bastante complexo quando tratamos de múltiplos objetivos, negação de objetivos e também com estados de causa e efeito não observados.

Dos experimentos realizados com agentes controlados por METs, observamos que, através de um processamento eficiente de objetivos, podemos obter agentes capazes de realizar diversos tipos de tarefas com um esforço mínimo de programação (que consiste na definição das estruturas hierárquicas dos metaesquemas) seguido de um processo de treinamento.

Consideremos inicialmente um metaesquema bastante simples que observa os seguintes espaços de causa e efeito:

- espaço de causa composto de seis estados: *A, B, C, D, E e F*;
- espaço de efeito composto de três estados: *X, Y e Z*.

Após um treinamento supervisionado, os esquemas de transição são apresentados na Tabela 8. Analisando esses esquemas, podemos obter um “mapa” das transições do espaço de efeito, conforme mostrado na Fig. 14.

TABELA 8
TRANSIÇÕES OBSERVADAS NO PROCESSO DE TREINAMENTO

Espaço de Causa		Espaço de Efeito		Tempo	Ocorrências	Possibilidade
Causa Inicial	Causa Final	Efeito Inicial	Efeito Final			
<i>Y</i>	<i>X</i>	<i>A</i>	<i>A</i>	-	3	-
<i>X</i>	<i>Y</i>	<i>A</i>	<i>B</i>	0,10	18	100%
<i>Y</i>	<i>X</i>	<i>B</i>	<i>B</i>	-	3	-
<i>X</i>	<i>Y</i>	<i>B</i>	<i>C</i>	0,23	6	80%
<i>X</i>	<i>Z</i>	<i>B</i>	<i>D</i>	0,05	12	50%
<i>Y</i>	<i>X</i>	<i>C</i>	<i>C</i>	-	7	-
<i>Z</i>	<i>X</i>	<i>C</i>	<i>C</i>	-	7	-
<i>X</i>	<i>Y</i>	<i>C</i>	<i>E</i>	0,12	5	90%
<i>X</i>	<i>Z</i>	<i>C</i>	<i>F</i>	0,15	1	100%
<i>X</i>	<i>X</i>	<i>D</i>	<i>A</i>	1,25	5	-
<i>X</i>	<i>Z</i>	<i>E</i>	<i>A</i>	0,15	5	70%
<i>Z</i>	<i>X</i>	<i>E</i>	<i>E</i>	-	5	-

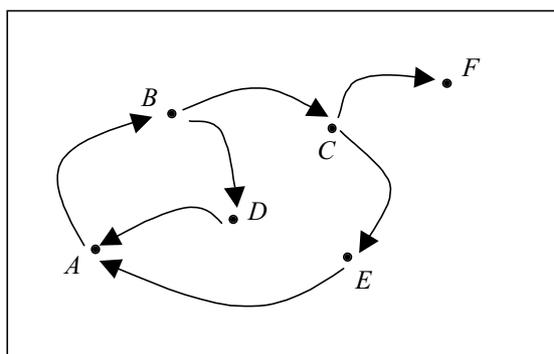


Fig. 14. Representação gráfica do espaço de efeito dado pela Tabela 8.

Uma análise preliminar da Tabela 8 e da Fig. 14 mostra que, em princípio, todos os estados do espaço de efeito podem ser percorridos através de transições no espaço de causa, porém, quando o estado *F* é atingido, os demais ficam inacessíveis. A transição do estado *D* para o estado *A* acontece de forma “aleatória” (transição sem causa observável) após tempo médio de permanência igual a 1,25 segundos.

Analisando o espaço de causa, verificamos que as transições *X-Y* e *X-Z* são as únicas que têm efeitos observáveis.

Para ilustrar o mecanismo básico de controle de METs, consideremos que, com base na Tabela 8, desejemos definir um único objetivo de saída, sendo dadas as seguintes entradas:

Caso 1:

- Estado de efeito = B
- Estado de causa = Z
- Entrada de objetivo = $\{A\}$

Da Fig. 14, observamos que existem dois “caminhos” partindo de B e chegando em A : o caminho $B-D-A$ e o caminho $B-C-E-A$. O primeiro caminho, apesar de mais curto, depende de uma causa não observada e, dessa forma, irá levar mais tempo do que o segundo. Assim, definimos um “objetivo imediato” que consiste em atingir o estado C . A Tabela 8 mostra que uma transição $X-Y$ deve gerar uma transição $B-C$ com uma possibilidade de 80% e um tempo médio de 0,23 segundos. Dessa forma, definimos como objetivo de saída o estado X , pois, em princípio, para que a transição $X-Y$ seja efetivada, o estado do espaço de causa deve assumir esse valor.

Caso 2:

- Estado de efeito = B
- Estado de causa = X
- Entrada de objetivo = $\{A\}$

Partindo da análise do caso anterior, definimos, como objetivo de saída, o estado Y para que a transição $X-Y$ seja efetivada.

Caso 3:

- Estado de efeito = C
- Estado de causa = Y
- Entrada de objetivo = $\{A\}$

Nesta condição, o objetivo imediato passa a ser o estado E e o objetivo de saída passa a ser o estado X , a fim de gerar uma nova transição $X-Y$.

Estes casos ilustram a evolução esperada para os estados de causa e, conseqüentemente, para os estados de efeito. Quando o estado definido na entrada de objetivos é atingido, o metaesquema fica bloqueado e um objetivo de saída nulo (ou vazio) é gerado.

Por exemplo, através das transições da Tabela 8, podemos concluir que, partindo do estado C , o estado objetivo A será atingido através de uma seqüência de quatro

objetivos de saída $\{X, Y, X, Z\}$. Se ocorrer alguma “falha”, essa seqüência deverá ser automaticamente corrigida. Existem alguns tipos de falhas que devem ser consideradas:

- a transição de causa não gerou a transição de efeito;
- uma transição sem causa observada mudou o estado do espaço de efeito;
- o espaço de causa não atingiu estado definido como objetivo de saída;
- o estado de causa variou para um estado diferente do objetivo de saída.

A primeira falha acima listada irá ocorrer algumas vezes sempre que o valor de possibilidade for inferior a 100%. Nesses casos, a transição de causa deve ser repetida se, após um tempo (por exemplo, duas vezes o tempo médio de transição), a transição de efeito não for observada.

A segunda falha listada é tratada de forma automática, pois uma mudança no espaço de efeito deve gerar automaticamente uma redefinição do “objetivo imediato”.

As últimas falhas estão relacionadas com problemas no próximo metaesquema da hierarquia, que processa os objetivos de saída (que são seus objetivos de entrada). Nesses casos, devemos esperar que o objetivo de saída seja atingido, mas, após um determinado tempo, podemos reconsiderar nosso “objetivo imediato” para algum outro que não dependa desse estado de causa “problemático”.

4.20 Tratamento de Objetivos Múltiplos

De forma genérica, os objetivos são conjuntos de valores pretendidos para os espaços de causa e efeito. Normalmente, trabalhamos com múltiplos objetivos de entrada e um único objetivo de saída. A implementação de múltiplos objetivos de saída leva a uma estrutura de controle muito mais complexa que não está sendo abordada neste trabalho.

Sempre que lidamos com objetivos, devemos ter em mente que tratamos com um processo de comunicação entre duas estruturas (normalmente, dois metaesquemas), no qual o objetivo de saída de uma é o objetivo de entrada de outra. Essa comunicação é, em princípio, unilateral com uma estrutura enviando objetivos para a outra. Esquemas mais elaborados poderiam se utilizar de uma comunicação bilateral, na qual, por exemplo, uma estrutura geraria um objetivo e a outra informaria sobre eventuais problemas em atingir esse objetivo, e outras, informações que estão disponíveis apenas no seu nível hierárquico. Nada impede que diversos metaesquemas gerem objetivos de saída referentes a um mesmo

sinal. Nesse caso, o metaesquema “responsável” por esse objetivo deverá receber simultaneamente dados de todos esses metaesquemas.

Quando lidamos com objetivos de entrada simultâneos e somente um objetivo de saída, esses objetivos podem ser priorizados de várias formas:

- ordem de chegada com fila tipo “primeiro a entrar, primeiro a sair”;
- ordem de chegada com fila tipo “primeiro a entrar, último a sair”;
- valor numérico de prioridade definido por quem gerou o objetivo;
- prioridade para o objetivo mais próximo, com maior possibilidade ou com menor tempo.

Sempre que um estado objetivo é atingido, ele é removido do conjunto. Quando o objetivo for nulo (conjunto vazio), o metaesquema fica bloqueado. Em uma estratégia de controle mais sofisticada, os metaesquemas podem gerar um “cancelamento” de um objetivo anteriormente definido. Esse cancelamento é usado para desativar objetivos de saída quando os objetivos que estão “acima” já foram atingidos. Essa não é uma situação muito freqüente, mas pode causar transtornos como, por exemplo, “passar do objetivo”, devendo, dessa forma, ser evitada através de um cancelamento.

4.21 Negação de Objetivos

Os “objetivos negados”, ou “anti-objetivos”, representam estados que devem ser evitados. Os sinais de reforço negativo são propagados normalmente através de objetivos negados. Dessa forma, esses sinais são um fator importante na implementação de estruturas de controle de agentes cognitivos. De forma geral, devemos considerar que um metaesquema contém duas listas: uma com os “objetivos” e outra com os “objetivos negados”, uma vez que os tratamentos de ambos são bastante distintos. Nas estruturas que implementamos, utilizamos apenas entradas de “objetivos negados” e nunca saídas, pois o processo de propagação de “objetivos negados” é bastante complexo. Os “objetivos negados” de entrada são convertidos em “objetivos” e, dessa forma, concorrem com os objetivos “normais”.

Podemos considerar que um “objetivo negado” é um estado que deve ser evitado pelo metaesquema. Isso pode ser feito de várias formas:

- forma passiva - não deixar que um “objetivo negado” seja considerado um “objetivo imediato”, assim o metaesquema deve evitar caminhos que conttenham “objetivos negados”, contornando esses caminhos;
- forma ativa indireta - estando em um estado em que um “objetivo negado” pode ser atingido por uma transição sem causa observável, devemos ter como “objetivo imediato” qualquer outro estado que se afaste do “objetivo negado”;
- forma ativa direta - estando em um estado representado por um “objetivo negado”, definir um “objetivo imediato” que possibilite afastar-se desse estado.

Algumas vezes devemos optar por realizar uma ação que nos leve a um objetivo ou realizar uma ação que evite um “objetivo negado”. Essa escolha não é genérica e depende do tipo de problema a ser tratado. Por exemplo, em uma partida do “Jogo da Velha”, o agente deve claramente optar por “vencer” em vez de bloquear o adversário. Em outras situações, o agente deve ter como prioridade não atingir um “objetivo negado”, como no caso em que o agente deve optar entre “atravessar a rua” e “não ser atropelado”.

Normalmente o objetivo de “não sentir dor” é mais importante do que o de “sentir prazer”, sendo que esse tipo de priorização pode ser informado ao agente ou ele pode aprender no seu contato com o meio ambiente.

4.22 Escolha de Caminhos

Normalmente, existem vários “caminhos” que podem ser escolhidos no processo de geração de “objetivos imediatos”, existindo diversos pontos que podem ser considerados nessa escolha:

- distância (número de estados) a ser percorrida;
- possibilidade cumulativa (produto das possibilidades de cada transição) do caminho;
- tempo médio total do caminho;
- dependência de transições sem causa observada para completar o caminho;
- existência de transições sem causa observada que possam causar um desvio;
- existência de “objetivos negados” no caminho ou “próximo” dele;
- existência de estados de causa “difíceis” de alcançar.

A consideração parcial ou total destas condições leva a algoritmos de controle de diferentes graus de complexidade e diferentes graus de desempenho. De forma geral, para tratar problemas genéricos, devemos adotar estratégias de controle mais abrangentes, no entanto, por vezes, soluções bastante simples geram bons resultados.

TABELA 9
PSEUDOCÓDIGO PARA DEFINIÇÃO DE OBJETIVOS DE SAÍDA DE UM MET

<p>Repetir Se (<i>EntradaObjetivo</i> <> <i>EntradaEfeito</i>) e (<i>EntradaObjetivo</i> <> <i>ValorNulo</i>) Então Início <i>PossibilidadeObjetivo</i> [<i>EntradaObjetivo</i>] = 1 Para cada <i>EsquemaDeTransição</i>[<i>i</i>] fazer Se <i>PossibilidadeObjetivo</i> [<i>EfeitoFinal</i>] > 0 Então <i>PossibilidadeObjetivo</i>[<i>EfeitoInicial</i>] = <i>PossibilidadeObjetivo</i>[<i>EfeitoFinal</i>] * <i>Possibilidade</i> Se <i>PossibilidadeObjetivo</i> [<i>EntradaEfeito</i>] > 0 Então Para cada <i>EsquemaDeTransição</i>[<i>i</i>] Fazer Se <i>EfeitoFinal</i> = <i>EntradaEfeito</i> Então Incluir este <i>EsquemaDeTrasnsição</i>[<i>i</i>] numa lista de sorteio Repetir <i>T</i> = <i>LerTempo</i> Sortear <i>ObjetivoImediato</i> da lista Para <i>EsquemaDeTransição</i>[<i>ObjetivoImediato</i>] Fazer Repetir Se <i>EntradaCausa</i> <> <i>CausaInicial</i> Então <i>SaídaObjetivo</i> = <i>CausaInicial</i> Senão <i>SaídaObjetivo</i> = <i>CausaFinal</i> <i>DT</i> = <i>Lertempo</i> - <i>T</i> Até (<i>EntradaEfeito</i> = <i>EfeitoFinal</i>) ou (<i>DT</i> > <i>TempoMedio</i> * 10) Até (<i>EntradaEfeito</i> = <i>EfeitoFinal</i>) ou (<i>EntradaEfeito</i> = <i>EntradaObjetivo</i>) Senão <i>SaídaObjetivo</i> = <i>ValorNulo</i> FimSe Até <i>FimDoPrograma</i></p>
--

Nos algoritmos de controle implementados neste trabalho, realizamos uma propagação inversa (partindo do estado final para o inicial) das possibilidades de transição, de modo a atribuir um valor unitário ao “estado objetivo”, calculando um índice para cada

“estado” até chegar ao estado atual. Nesse caso, o estado que levasse a um maior índice seria definido como “objetivo imediato”.

Uma estratégia interessante é sortear um “objetivo imediato” dentre aqueles que apresentaram maior índice de possibilidade (dando maiores chances aos que são considerados melhores). Assim, geramos uma certa “aleatoriedade” nas ações do agente quando este se depara com “caminhos” equivalentes, o que irá contribuir com os mecanismos de exploração do ambiente. Se o metaesquema escolher sempre um mesmo caminho que considera o melhor, não irá perceber possíveis mudanças no ambiente que tenha privilegiado caminhos, que anteriormente foram considerados ruins. A Tabela 9 apresenta um algoritmo básico no qual são consideradas as possibilidades de transição no sorteio de um “objetivo imediato”. Esse algoritmo é bastante simples. Nele, um algoritmo um pouco mais elaborado poderia “perguntar” para o próximo metaesquema: - qual é a possibilidade de atingir um dado objetivo de saída visando principalmente identificar objetivos que não possam ser atingidos e evitando uma perda de tempo gerada pela espera desse objetivo?

Quando nenhum “caminho” estiver disponível para que um determinado estado seja alcançado, um objetivo de saída nulo deve ser gerado. Essa situação irá ocorrer, principalmente, no início do processo de treinamento, quando os espaços de causa e efeito ainda não estão completamente mapeados. De forma análoga, para esquemas sem causa observável, um objetivo de saída nulo também deve ser adotado a fim de esperar que uma transição ocorra espontaneamente.

4.23 Definição de Objetivos de Saída

Um “objetivo imediato” irá sempre derivar de um estado de efeito que pode ser atingido com uma única transição no espaço de efeito que estará associada a uma transição no espaço de causa. Dessa forma, o objetivo de saída deve ser definido de forma que a transição no espaço de causa venha a ocorrer, gerando, assim, a transição desejada para que o “objetivo imediato” seja atingido. Ao gerar um objetivo de saída, o metaesquema “quebra” um objetivo de entrada em novos objetivos mais específicos que são imediatamente repassados a outros “metaesquemas”.

Um objetivo de saída só se transforma em ação quando é definido sobre um sinal

de saída do sistema. Dessa forma, um objetivo inicial, como, por exemplo, “elevar a temperatura”, pode gerar uma série de novos objetivos, tais como: “diminuir a vazão da água”, “fechar a torneira” e, por fim, “colocar a mão na torneira e girá-la no sentido anti-horário”. Esse último objetivo, por sua vez, pode ser dividido em “abrir a mão”, “levar a mão até a torneira”, “fechar a mão” e “girar a mão”. Em um agente muito simples a função de “fechar a mão” pode ser diretamente implementada por um único sinal de saída (que é ativado quando o objetivo “fechar a mão” for identificado). Em agentes mais complexos, poderíamos ter metaesquemas que definissem saídas ainda mais básicas, como, por exemplo, o movimento de cada “dedo” ou de cada “músculo” individual.

É importante observar que, para transmissão de objetivos entre estados que são relacionados com funções de transferência bem definidas, podemos utilizar estruturas de controle tradicionais. Dessa forma, um objetivo do tipo “fechar a mão” pode acionar uma rotina de controle tradicional que defina valores para um sistema complexo de acionamento de uma mão robotizada.

Um outro ponto importante no processo de definição de um objetivo de saída é saber com que tipo de transição de causa estamos lidando:

- para transições de causa convergente, apenas o estado final de causa deve ser definido como objetivo de saída. No caso desse estado já estar “ativado”, devemos primeiro definir um outro estado qualquer e, a seguir, retornar ao estado desejado;
- para transições de causa divergente, o estado inicial deve ser definido e, a seguir, um outro estado qualquer pode ser escolhido;
- para transições completas, os dois estados da transição de causa devem ser definidos como objetivos de saída.

Poderíamos considerar que todas transições podem ser tratadas como transições completas, mas essa abordagem é muito restritiva, pois limita o percurso dos objetivos de saída a um único caminho, que, em alguns casos, poderia estar bloqueado. Um bom exemplo de transição de causa convergente são os jogos, como o “Jogo da Velha”, em que a seqüência de movimentos não é tão importante como a posição final das peças. Se, nesse exemplo, utilizarmos apenas transições completas, limitaremos em muito a capacidade do agente.

A classificação de tipo de transição de causa pode ser feita pelo próprio agente após a observação de um grande conjunto de transições. Como essa análise leva a um maior esforço de treinamento nos experimentos do presente trabalho, optamos por informar ao metaesquema qual o tipo de transição de causa com que ele iria lidar.

Um aspecto importante da definição de objetivos é o caso no qual em um mesmo objetivo de saída tenhamos “misturados” sinais de saída do agente com sinais objetivos que serão tratados por outros metaesquemas. Nesses casos, as saídas só poderão ser ativadas quando todos os objetivos forem alcançados. Para exemplificar, consideremos um meta esquema que receba o objetivo de entrada “ Pegar um copo” e gere o objetivo de saída “Aproximar a mão do copo; Fechar a mão”. Considerando-se que o objetivo “Aproximar a mão do copo” levará algum tempo para ser atingido (pois será enviado a outro metaesquema) e o objetivo “Fechar a mão” aciona diretamente uma saída, se não esperarmos o primeiro objetivo ser cumprido, a mão será fechada antes de chegar ao “copo”.

4.24 Considerações sobre Estruturas Baseadas em Metaesquemas de Transição

Os metaesquemas de transição estudados neste trabalho foram aplicados ao controle de agentes cognitivos inseridos em um mundo artificial. Esses agentes foram treinados através de sinais de reforço positivo e negativo em uma abordagem totalmente diversa da normalmente utilizada, pois esses sinais foram empregados para definir o objetivo equivalente a “sentir prazer”, e o “objetivo negado” equivalente a “sentir dor”. Através das estruturas de metaesquemas, esses objetivos são “quebrados” e propagados de forma hierárquica até chegarem a ações básicas.

Como já mencionamos, o processamento de um metaesquema depende de três procedimentos que são executados continuamente dentro do metaesquema:

- mecanismos de exploração;
- procedimentos de manutenção dos esquemas;
- procedimentos de definição de saídas objetivos.

Cada metaesquema executa estes procedimentos de forma independente,

comunicando-se com outros metaesquemas apenas para receber e transmitir objetivos. Em princípio, nenhum sinal de sincronismo é necessário aos metaesquemas, mesmos no caso da realização de tarefas complexas que demandam seqüências de ações.

Nesta seção, apresentamos algumas considerações básicas sobre a implementação destes procedimentos, porém temas importantes, como a criação de sinais internos para tratamento de estados não observados, ainda estão em aberto. De forma bastante rápida, vislumbramos diversos tipos de estratégias e heurísticas que poderiam ser utilizados no processamento de metaesquemas, principalmente na definição de objetivos de saída. Assim, temos plena certeza de que muito trabalho de pesquisa ainda deva ser realizado para uma definição de estratégias otimizadas para processamento de metaesquemas. Além disso, os próprios processos de criação e implementação de estrutura baseadas em METs abrem inúmeras possibilidades que certamente não serão tão cedo esgotadas.

4.25 Resultados Experimentais

O modelo de controle de objetivos proposto foi validado de forma experimental através do treinamento condicionado de diversos agentes autônomos. Com base no modelo de controle apresentado, definimos uma máquina de inferência para controle de metaesquemas de transição para controle do agente. Além disso, foram definidos uma série de simuladores com níveis de complexidade crescente a fim de validar os conceitos propostos. Foram tratados os seguintes casos:

- problema da máquina de caça-níqueis;
- controle de agentes em diversos ambientes;
- controle do braço de um robô;
- problema do “Jogo da Velha”.

Para todos estas implementações, o agente foi capaz de realizar as tarefas propostas através de um processo contínuo de aprendizagem. Para cada caso, o agente foi “programado” através da interligação de uma hierarquia de metaesquemas, utilizando-se a sintaxe de programação definida na Seção 4.11. Em todos os casos, a mesma máquina de inferência foi utilizada para o controle dos metaesquemas. Neste capítulo, apresentaremos apenas a implementação do controle do braço de um robô, sendo as demais implementações abordadas no Apêndice 1.

4.26 Controle do Braço de um Robô

Este problema consiste em um agente cognitivo relativamente complexo controlado por uma série de METs distintos. O agente possui um sistema sensorial simples com três tipos de percepções (visão, tato e paladar) e um sistema motor, realizando ações de deslocamento em uma esteira na qual se encontram objetos e, ainda, controlando uma mão e uma boca.

A Fig. 15 mostra o simulador definido para este problema, como também as entradas e saídas do agente. Uma série de botões permitem um conjunto de 7 ações (abrir a mão, fechar a mão, subir a mão, descer a mão, abrir a boca, fechar a boca e mover a esteira).

Os objetos são “doces” e “amargos”, mas o agente só percebe essa qualidade ao colocá-los na boca. Ele recebe um reforço positivo ao “comer” os objetos doces e um reforço negativo ao “comer” os objetos amargos.

Existem, no sistema, uma série de restrições do tipo:

- a esteira não anda, se a mão estiver abaixada;
- o sensor de tato só atua com a mão fechada;
- o sensor de paladar só atua com o objeto na boca e fora da mão;
- ao abrir a mão, o objeto cai da mão;
- ao abrir a mão com o objeto na boca, ele fica na boca;
- a boca só pode ser aberta e fechada, se a mão não estiver sobre a boca.

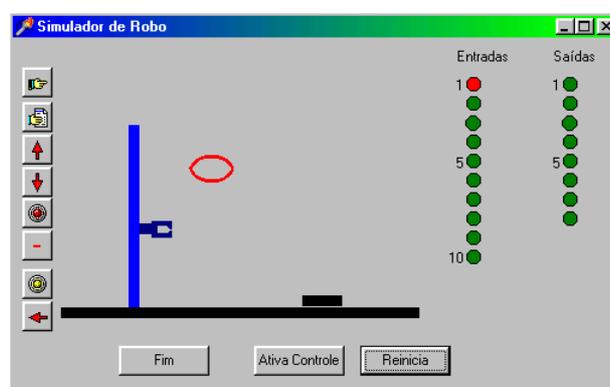


Fig. 15. Tela do simulador do agente definido neste problema.

O agente possui dez entradas:

E₁ - estado da mão (1=fechada);

E₂ e E₃ - posição da mão (01=embaixo, 10=em cima);

E₄ - sensor de tato (1=segurando objeto);

E₅ - estado da boca (1=fechada);

E₆ e E₇ - sensor de paladar (01=doce, 10=amargo);

E₈ - visão (1=objeto visualizado);

E₉ e E₁₀ - sensores de reforço (01=prazer,10=dor).

O agente também possui as seguintes saídas:

S₁ - abre a mão;

S₂ - fecha a mão;

S₃ - sobe a mão;

S₄ - desce a mão;

S₅ - abre a boca;

S₆ - fecha a boca;

S₇ - move a esteira em um passo.

O programa de controle definido para este problema é composto de 7 METs, conforme mostrado na Tabela 10.

- O primeiro MET irá relacionar os sinais de reforço com o estado da boca, gosto do objeto, conforme mostra a Tabela 11.
- O segundo MET irá relacionar o objetivo de “sentir o gosto do objeto” com a posição da mão, o tato e o estado da boca, conforme mostra a Tabela 12.
- O terceiro MET irá relacionar o objetivo “sentir que está segurando um objeto” com o estado da mão, a posição da mão e a visão de um objeto, conforme mostra a Tabela 13.
- O quarto MET irá relacionar o objetivo “ver objeto” com o movimento da esteira e a posição da mão, conforme mostra a Tabela 14.
- O quinto MET irá relacionar o estado da boca com a posição da mão e com as ações de abrir e fechar a boca, conforme mostra a Tabela 15.
- O sexto MET irá relacionar o estado da mão com as ações de abrir e fechar a mão, conforme mostra a Tabela 16.

- O sétimo MET irá relacionar a posição da mão com as ações de subir e descer a mão, conforme mostra a Tabela 17.

TABELA 10
PROGRAMA DE CONTROLE DO AGENTE

<p>% Programa de controle do braço robotizado</p> <p>% MET 1</p> <p>% Espaço de causa = estado da boca, gosto do objeto</p> <p>% Espaço de efeito = prazer, dor</p> <p>(E5, E6, E7 E9, E10 O5, O6, O7 1, 0 0,1)</p> <p>% MET 2</p> <p>% Espaço de causa = posição da mão, tato, estado da boca</p> <p>% Espaço de efeito =gosto</p> <p>(E2, E3, E4, E5 E6, E7 O2,O3, O4, O5 O6, O7)</p> <p>% MET 3</p> <p>% Espaço de causa = estado da mão, posição da mão, ver objeto, gosto do objeto</p> <p>% Espaço de efeito = tato</p> <p>(E1, E2, E3, E8, E6, E7 E4 O1, O2, O3, O8 O4)</p> <p>% MET 4</p> <p>% Espaço de causa = move a esteira</p> <p>% Espaço de efeito = vê objeto:</p> <p>(S7, E2, E3 E8 S7, O2, O3 O8)</p> <p>% MET 5</p> <p>% Espaço de causa = abre e fecha a boca, posição da mão</p> <p>% Espaço de efeito = estado da boca</p> <p>(S5, S6, E2, E3 E5 S5, S6, O2, O3 O5)</p> <p>% MET 6</p> <p>% Espaço de causa = sobe e desce a mão</p> <p>% Espaço de efeito = posição da mão:</p> <p>(S3, S4 E2, E3 S3, S4 O2, O3)</p> <p>% MET 7</p> <p>% Espaço de causa = abre e fecha a mão</p> <p>% Espaço de efeito = estado da mão</p> <p>(S1, S2 E1 S1, S2 O1)</p> <p>FIM</p>
--

A seguir são apresentadas as tabelas de transição dos sete METs utilizados no controle do agente.

TABELA 11
TRANSIÇÕES OBSERVADAS PELO PRIMEIRO MET

Espaço de Causa (boca fechada, doce, amargo)		Espaço de Efeito Inicial (dor, prazer)		Possibilidade
Causa Inicial	Causa Final	Efeito Inicial	Efeito Final	
0, 1, 0	1, 0, 0	0, 0	0, 1	1,00
0, 0, 1	1, 0, 0	0, 0	1, 0	1,00
1, 0, 0	1, 0, 0	0, 1	0, 0	1,00
1, 0, 0	1, 0, 0	1, 0	0, 0	1,00

TABELA 12
TRANSIÇÕES OBSERVADAS PELO SEGUNDO MET

Espaço de Causa (mão no chão, mão na boca, segurando objeto, boca fechada)		Espaço de Efeito Inicial (doce, amargo)		Possibilidade
Causa Inicial	Causa Final	Efeito Inicial	Efeito Final	
0, 1, 1, 0	0, 1, 0, 0	0, 0	1, 0	0,55
0, 1, 1, 0	0, 1, 0, 0	0, 0	0, 1	0,45
0, 1, 0, 0	0, 1, 0, 0	0, 1	0, 0	1,00
0, 1, 0, 0	0, 1, 0, 0	1, 0	0, 0	1,00
0, 0, 0, 0	0, 0, 0, 0	1, 0	0, 0	1,00
0, 1, 0, 0	0, 1, 1, 0	1, 0	0, 0	1,00
0, 1, 0, 0	0, 1, 1, 0	0, 1	0, 0	1,00

TABELA 13
TRANSIÇÕES OBSERVADAS PELO TERCEIRO MET

Espaço de Causa (mão fechada, mão no chão, mão na boca, vendo objeto, doce, amargo)		Espaço de Efeito Inicial (segurando objeto)		Possibilidade
Causa Inicial	Causa Final	Efeito Inicial	Efeito Final	
0, 1, 0, 1, 0, 0	1, 1, 0, 1, 0, 0	0	1	1,00
1, 0, 1, 0, 0, 0	0, 0, 1, 0, 0, 0	1	0	1,00
0, 0, 1, 0, 1, 0	1, 0, 1, 0, 1, 0	0	1	1,00
0, 0, 1, 0, 0, 1	1, 0, 1, 0, 0, 1	0	1	1,00

TABELA 14
TRANSIÇÕES OBSERVADAS PELO QUARTO MET

Espaço de Causa (mover esteira, mão no chão, mão na boca)		Espaço de Efeito Inicial (vendo objeto)		Possibilidade
Causa Inicial	Causa Final	Efeito Inicial	Efeito Final	
0, 0, 0	1, 0, 0	0	1	0,23
0, 0, 1	1, 0, 1	0	1	0,23
0, 1, 0	0, 1, 0	1	0	1,00
0, 0, 1	1, 0, 1	1	0	1,00

TABELA 15
TRANSIÇÕES OBSERVADAS PELO QUINTO MET

Espaço de Causa (abre a boca, fecha boca, mão no chão, mão na boca)		Espaço de Efeito Inicial (boca fechada)		Possibilidade
Causa Inicial	Causa Final	Efeito Inicial	Efeito Final	
0, 0, 0, 0	0, 1, 0, 0	0	1	1,00
0, 0, 1, 0	0, 1, 1, 0	0	1	1,00
0, 0, 0, 0	1, 0, 0, 0	1	0	1,00
0, 0, 1, 0	1, 0, 1, 0	1	0	1,00

TABELA 16
TRANSIÇÕES OBSERVADAS PELO SEXTO MET

Espaço de Causa (sobe mão, desce a mão)		Espaço de Efeito Inicial (mão no chão, mão na boca)		Possibilidade
Causa Inicial	Causa Final	Efeito Inicial	Efeito Final	
0, 0	0, 1	0, 1	0, 0	1,00
0, 0	0, 1	0, 0	1, 0	1,00
0, 0	1, 0	1, 0	0, 0	1,00
0, 0	1, 0	0, 0	0, 1	1,00

TABELA 17
TRANSIÇÕES OBSERVADAS PELO SÉTIMO MET

Espaço de Causa (abre a mão, fecha a mão)		Espaço de Efeito Inicial (mão fechada)		Possibilidade
Causa Inicial	Causa Final	Efeito Inicial	Efeito Final	
0, 0	0, 1	0	1	1,00
0, 0	1, 0	1	0	1,00

Uma descrição passo a passo da forma de operação do conjunto de metaesquemas, que controlam o agente, é dada a seguir:

- o objetivo “sentir prazer” dispara o objetivo “sentir gosto doce e fechar a boca”;
- o objetivo “sentir gosto doce” dispara o objetivo “sentir tato, mão em cima, mão aberta, boca aberta”;
- o objetivo “sentir tato” dispara o objetivo “ver objeto, mão embaixo, mão fechada”;
- o objetivo “ver objeto” dispara o objetivo “mover esteira, mão acima da esteira”;
- se a mão estiver embaixo ($E_2 = 0$ e $E_3 = 1$), o objetivo “mão acima da esteira” gera o objetivo “subir a mão”;
- o objetivo “subir a mão” ativa a saída S_3 ;

- após a mão subir, o objetivo “mover esteira” ativa a saída S_7 ;
- após um tempo, se um objeto não for detectado, a saída S_7 é ativada novamente;
- após ver um objeto, o “sentir tato” dispara o objetivo “mão embaixo, mão fechada”;
- o objetivo “mão embaixo” ativa a saída S_4 ;
- o objetivo “mão fechada” primeiro abre a mão (ativa a saída S_1) e depois fecha a mão (ativa a saída S_2);
- abrindo e fechando a mão sobre o objeto, este é pegado pela mão e o sensor de tato é ativado;
- após ativar o sensor de tato, o objetivo “sentir gosto” dispara os objetivos “mão em cima, mão aberta, boca aberta”;
- o objetivo “ boca aberta” ativa a saída S_5 ;
- o objetivo “mão em cima” ativa duas vezes a saída S_3 ;
- o objetivo “mão aberta” ativa saída S_1 ;
- abrindo a mão sobre a boca aberta, o agente sente o gosto do objeto;
- se o gosto for doce, o objetivo “fechar boca” é ativado;
- o objetivo “fechar boca” gera o objetivo “descer a mão”;
- o objetivo “descer a mão” ativa a saída S_4 ;
- O objetivo “fechar boca” ativa a saída S_6 ;
- com estas ações, o agente come um objeto doce e recebe um sinal de reforço positivo.

Para executar este tipo de tarefa, o agente utiliza em média 12 ações, o que para as 7 saídas disponíveis gera um espaço com aproximadamente 13×10^9 estados, tornando praticamente inviáveis as abordagens tradicionais de treinamento com reforço.

Um problema inerente à estrutura de processamento de METs utilizada é a geração de objetivos contraditórios definidos por dois METs (por exemplo, um manda abrir a mão e outro manda fechá-la). Isso pode ser facilmente resolvido através dos seguintes mecanismos:

- os objetivos dos METs que estão mais baixos na estrutura são prioritários;
- os últimos objetivos recebidos são prioritários.

Este foi o problema mais complexo que resolvemos através da aplicação de METs, porém não vemos muitas limitações para o tratamento de qualquer problema real. Fora os comentários, o “programa” de controle deste agente tem apenas 7 linhas de instruções mostradas na Tabela 18 (se utilizarmos uma interface gráfica, teremos sete blocos sendo interligados), o que para nós, por si só, representa um grande avanço, se considerarmos que um pseudocódigo que realize a mesma função (para um ambiente estático) deva consumir de 20 a 30 linhas de programa.

TABELA 18
PROGRAMA DE CONTROLE DO AGENTE

(E4, E6, E7 E9, E10 O6, O7 1, 0 0,1)
(E2, E3, E4, E5 E6, E7 O2,O3, O4, O5 O6, O7)
(E1, E2, E3, E8, E6, E7 E4 O1, O2, O3, O8 O4)
(S7, E2, E3 E8 S7, O2, O3 O8)
(S5, S6, E2, E3 E5 S5, S6, O2, O3 O5)
(S3, S4 E2, E3 S3, S4 O2, O3)
(S1, S2 E1 S1, S2 O1)
FIM

4.27 Conclusões

O modelo de controle aqui apresentado representa um novo paradigma que é uma evolução dos conceitos de controle tradicional baseado no modelo “entrada - processamento - saída”. Como já enfatizamos, o modelo proposto não diferencia “sinais de entrada” de “sinais de saída”, sendo baseado em relações de causa e efeito representadas por meio de esquemas de transição. O processo de aprendizagem dos agentes no modelo proposto assemelha-se muito ao observado em seres humanos, o que de certa forma era esperado, já que a base teórica utilizada está fortemente apoiada nos conceitos de esquemas formulados por Piaget no contexto da psicologia humana.

O modelo proposto é bastante genérico e de fácil programação, tendo sido aplicado com sucesso a alguns problemas de controle que, apesar de simples, são bastante representativos:

- para o caso do controle do braço de um robô, o agente aprendeu a realizar uma série de ações que dificilmente poderiam ser aprendidas com técnicas de

treinamento tradicionais. Através de treinamento supervisionado, o agente chega a aprender uma tarefa em uma única etapa;

- para o caso do “Jogo da Velha”, os resultados foram surpreendentes, pois o agente aprende observando as táticas de seu adversário e as usa no seu jogo. Se observarmos, na Tabela 4, a quantidade de informação necessária para a “programação” do agente para implementar o “Jogo da Velha”, podemos concluir facilmente que ela é muito mais reduzida do que a requerida por qualquer técnica tradicional de programação.

As técnicas existentes de treinamento não supervisionado tradicionais têm uma certa dificuldade em tratar problemas mais complexos. Mesmo problemas como o do “Jogo da Velha” demandam um razoável esforço de treinamento, pois, nesse caso, o reforço é dado somente no fim do jogo e fica difícil saber em que ação específica um erro ou um acerto foi cometido. Uma análise mais profunda mostra que o problema não está nas técnicas, que são muito similares, mas sim no modelo de controle do agente, representado por uma única função que define o valor das saídas a partir do valor das entradas. Em princípio, dentro de uma visão reducionista, podemos considerar que existe sempre uma função de mapeamento que define uma única saída para cada entrada. Porém, para problemas práticos, essas funções serão tão complexas que dificilmente poderão ser implementadas.

Assim, para construir um programa que aprenda a jogar Xadrez através de treinamento com reforço, baseado nos modelos tradicionais, teremos uma série de problemas:

- o caminho para apresentar todas as jogadas possíveis, informando para cada uma delas qual o melhor movimento das peças, torna-se inviável por dois motivos: número proibitivo de jogadas possíveis e a impossibilidade de saber qual a melhor jogada para cada configuração;
- o caminho para jogar uma partida inteira e dar um reforço no final se torna inviável, pois é praticamente impossível distinguir, em uma seqüência de 20 ou 30 jogadas, quais foram corretas e quais não foram, partindo apenas do resultado final.

Desta forma, podemos afirmar que é praticamente impossível ensinar um agente

cognitivo a jogar Xadrez usando treinamento com reforço através do método tradicional. No entanto, podemos considerar essa dificuldade mais como uma deficiência do método tradicional do que uma característica do problema, pois os seres humanos são capazes de aprender desta forma, consistentemente ao modelo proposto.

Com o novo modelo, podemos definir uma série de metaesquemas. Um que analise os movimentos do nosso rei e, outro, os do rei adversário, e assim sucessivamente para cada uma das 32 peças. Cada metaesquema irá aprender a mover sua peça e saber se está no jogo ou não. Além disso, cada peça poderá identificar quando está ameaçada ou pode capturar uma peça adversária ou não pode se mover (está bloqueada). O objetivo máximo do jogo seria bloquear o rei adversário e o objetivo máximo negado seria deixar que nosso rei seja bloqueado. Metaesquemas um pouco mais abstratos que aprendam seqüências de abertura e formas de finalização poderão também ser implementados e treinados. Assim, o sistema será treinado de forma muito semelhante com a qual ensinaríamos um ser humano a jogar.

Neste contexto, podemos afirmar que o modelo proposto representa um grande avanço com relação aos existentes, pois permite aplicar técnicas de treinamento com reforço e também de treinamento não supervisionado em tarefas muito mais complexas. No treinamento de robôs, um instrutor pode assumir momentaneamente o controle das saídas e conduzir o robô em uma determinada tarefa, em uma postura muito semelhante àquela adotada quando se ensina uma criança a escrever ou a escovar os dentes, segurando e conduzindo suas mãos.

Assim, o modelo proposto representa um interessante passo na direção de sistemas de IA efetivamente inteligentes, que aprendem continuamente dentro do ambiente em que estão inseridos.

Memórias Associativas Holográficas

5.1 Introdução

O modelo de esquemas de transição proposto no capítulo anterior revelou-se bastante eficiente quando aplicado à realização de tarefas seqüenciais, mas não é diretamente aplicável a tarefas que requeiram memorização de grandes quantidades de informações, como é o caso dos problemas relacionados à interpretação de símbolos de linguagem. Buscando formas de aperfeiçoar o modelo proposto, verificamos que seria necessário utilizar um tipo de memória para armazenar seqüências de símbolos. Essa memória não poderia ser “endereçável”, mas deveria operar através da associação de informações, ou seja, deveria ser uma memória associativa. As memórias associativas encontradas na literatura apresentam uma série de limitações, sendo a estrutura proposta por Plate [45], denominada *holographic reduced representation* (HRR), uma das memórias mais eficientes que identificamos. Estudando a forma de operação e as vantagens e deficiências da HRR, idealizamos um novo tipo de memória que supera a HRR em alguns aspectos. A nova estrutura foi denominada Memória Associativa Holográfica (MAH) e foi utilizada para armazenamento e recuperação de informações simbólicas.

A Memória Associativa Holográfica emprega estruturas de dados baseadas em padrões randômicos para armazenar as informações desejadas. Essa forma de armazenamento dispensa qualquer processo de treinamento e não requer procedimentos complexos de restauração dos dados de saída. Como a informação está distribuída em toda a memória, partes do seu conteúdo podem ser corrompidas ou mesmo apagadas sem que informações relevantes sejam perdidas.

5.2 Memórias Associativas

Descreveremos, a seguir, os conceitos básicos das memórias associativas e suas formas de implementação, as mais usuais encontradas na literatura.

Memórias associativas [46] são estruturas que permitem o armazenamento e a restauração (recuperação) de informações com base na associação de padrões.

Diferentemente das memórias convencionais, que acessam seus dados através de endereços físicos predefinidos, as memórias associativas armazenam conjuntos de informação que podem ser representados (modelados) como pares de vetores $(\mathbf{x}_k, \mathbf{y}_k)$, definidos por:

$$\begin{aligned} \mathbf{x}_k \in \mathfrak{R}^n &\leftrightarrow \mathbf{x}_k = [x_{k1}, \dots, x_{kn}]^T, & x_{ki} \in \mathfrak{R}, \\ \mathbf{y}_k \in \mathfrak{R}^m &\leftrightarrow \mathbf{y}_k = [y_{k1}, \dots, y_{km}]^T, & y_{ki} \in \mathfrak{R}, \end{aligned} \quad (5.1)$$

onde \mathfrak{R}^n e \mathfrak{R}^m representam espaços reais de dimensões n e m , respectivamente, e $k = 1, \dots, p$ está caracterizando o número total de pares associados.

A memória associativa realiza uma transformação no espaço vetorial $T: \mathfrak{R}^n \rightarrow \mathfrak{R}^m$, mapeando o vetor \mathbf{x}_k no \mathbf{y}_k .

Se as dimensões dos vetores \mathbf{x}_k e \mathbf{y}_k forem iguais, a memória é denominada auto-associativa [47], se forem diferentes (caso geral), a memória é denominada heteroassociativa [48]. As memórias auto-associativas são usadas, geralmente, em aplicações de identificação de padrões [49]. Por outro lado, as memórias heteroassociativas são mais abrangentes em termos de aplicações. Como nessas memórias, o acesso pode ser realizado de forma recursiva, uma informação pode estar relacionada à outra, compondo uma cadeia de dados, de forma análoga ao que se presume ocorrer com a memória humana.

As memórias associativas vêm sendo amplamente empregadas em sistemas conexionistas (baseados em redes neurais artificiais (RNA)), visando, além do armazenamento, também o processamento e a representação de dados de forma distribuída. Uma aplicação típica nessa área são sistemas de processamento de linguagem natural [50].

As memórias associativas propostas na literatura têm ainda aplicação limitada,

apresentando alguns inconvenientes de ordem prática, tais como:

- necessitam de um processo de treinamento que, além de demandar normalmente um grande esforço computacional, dificultam a aplicação de tais memórias em sistemas operando em tempo real;
- para a maior parte das memórias, o dado recuperado (extraído) apresenta um alto nível de degradação, exigindo um procedimento complexo para a sua restauração.

Estes inconvenientes foram em grande parte superados no modelo de Memória Associativa Holográfica proposto neste trabalho.

5.3 Memórias Associativas Encontradas na Literatura

As memórias associativas podem ser implementadas basicamente de duas formas:

- ortogonalização e mapeamento: os vetores de entrada são ortogonalizados (ou classificados em grupos de padrões). Os vetores de saída são obtidos pela aplicação de uma função de mapeamento sobre os vetores de entrada ortogonalizados;
- transformações inversíveis: os pares de vetores de entrada e saída são combinados através de algum tipo de operação inversível, sendo todas as combinações acumuladas e representadas por um único vetor. Para a obtenção dos vetores primitivos armazenados, uma transformação inversa é aplicada.

Exemplos de memórias associativas encontradas na literatura:

5.3.1 Memória Associativa Linear (LAM)

Esta memória é composta de uma matriz \mathbf{M} que mapeia as entradas diretamente nas saídas:

$$\mathbf{y} = \mathbf{M}\mathbf{x} . \quad (5.2)$$

A matriz \mathbf{M} pode ser obtida diretamente da matriz de correlação entre os pares de entrada e saída [51]. Se os vetores de entrada forem ortogonais, os padrões de saída permitem a perfeita restauração dos padrões de entrada. Se os vetores não são ortogonais

(o que normalmente ocorre), teremos erros na saída causados pela interferência entre os padrões.

5.3.2 Memória Associativa de Segunda Ordem (SAM)

Neste tipo de memória, uma transformação não-linear é introduzida com o objetivo de reduzir o problema da não-ortogonalidade dos padrões de entrada, dessa forma:

$$y = \mathbf{M}T(x), \quad (5.3)$$

onde $T(x)$ é uma transformação não-linear (por exemplo, uma transformação polinomial) e a matriz \mathbf{M} é agora obtida via processo de otimização linear [52].

5.3.3 Memória Recursiva Auto-associativa (RAAM)

Em 1990, Pollack [53] apresentou uma estrutura de memória denominada *recursive auto-associative memory* (RAAM). Essa memória era obtida através de uma RNA de duas camadas, treinada via algoritmo *backpropagation*.

Na estrutura RAAM, dois vetores de dimensão n são apresentados a uma rede com $2n$ entradas, n neurônios na primeira camada e $2n$ neurônios na segunda camada. A rede é treinada para apresentar na saída os mesmos vetores de entrada. Após o treinamento, para um dado conjunto de vetores, observa-se que a primeira camada da rede realiza uma compactação dos dois vetores de entrada para um único vetor, enquanto a segunda camada realiza a descompactação. Dessa forma, a primeira camada de uma RAAM pode ser utilizada de forma recursiva a fim de armazenar uma seqüência de vetores em um único vetor. Essa seqüência pode ser restaurada através da segunda camada da RNA de forma também recursiva.

5.3.4 Memória Baseada em Convolução Circular (HRR)

Em 1994, Plate [54] propôs uma estrutura denominada *holographic reduced representation* (HRR). Essa abordagem era baseada em operações de convolução circular e correlação circular para o armazenamento e a restauração dos dados, respectivamente. A

convolução circular (representada por \otimes) é basicamente uma operação de convolução entre dois vetores, conservando a dimensão do vetor gerado. A correlação circular (representada por \oplus) é a operação inversa da convolução circular, podendo-se demonstrar que, para certos casos (vetores de grande dimensão não correlacionados), a Eq. (5.4) pode ser satisfeita:

$$(\mathbf{y}_k \otimes \mathbf{x}_k) \oplus \mathbf{x}_i = \begin{cases} \mathbf{y}_i, & k = i \\ 0, & k \neq i \end{cases} \quad (5.4)$$

Os pares de vetores são então armazenados através do procedimento descrito pela Equação (5.5):

$$\mathbf{s} = \mathbf{x}_1 \otimes \mathbf{y}_1 + \mathbf{x}_2 \otimes \mathbf{y}_2 + \cdots + \mathbf{x}_p \otimes \mathbf{y}_p. \quad (5.5)$$

Assim, para recuperar um dos vetores, basta aplicar (5.4) em (5.5). Portanto,

$$\begin{aligned} \mathbf{y}_i &= \mathbf{s} \oplus \mathbf{x}_i \\ \mathbf{x}_i &= \mathbf{s} \oplus \mathbf{y}_i \end{aligned} \quad (5.6)$$

5.4 Memória Associativa Holográfica

A estrutura de memória proposta, que foi denominada memória associativa holográfica [55], está baseada na utilização de uma transformação inversível, atuando de forma similar à memória HRR, sendo aplicável para vetores de entrada e saída representados em uma base binária.

Assim, seja $\mathbf{E}(k)$ um conjunto de matrizes retangulares, cujas linhas são formadas por vetores independentes normalizados (ortonormais) obtidos de uma base de números aleatórios. O produto de quaisquer duas dessas matrizes, denominadas matrizes randômicas, (sendo uma delas transposta) é dado por:

$$\mathbf{E}(k)\mathbf{E}^T(i) \cong \begin{cases} \mathbf{I}, & k = i \\ 0, & k \neq i \end{cases} \quad (5.7)$$

Consideremos agora que cada vetor de entrada (\mathbf{x}_k) possa estar associado a uma matriz ($\mathbf{E}_x(k)$), pertencente a um conjunto de matrizes randômicas que satisfazem a expressão (5.7). Assim,

$$\mathbf{x}_k \Rightarrow \mathbf{E}_x(k). \quad (5.8)$$

Consideremos também que cada vetor de saída (\mathbf{y}_k) possa ser associado a uma matriz (\mathbf{Y}_k) que carrega as mesmas informações desse vetor em sua diagonal principal. Assim,

$$\mathbf{y}_k \Leftrightarrow \mathbf{Y}_k. \quad (5.9)$$

A expressão que descreve a operação de armazenamento dos vetores de saída na memória associativa holográfica é dada por:

$$\mathbf{S} = \mathbf{Y}_1 \mathbf{E}_x(1) + \mathbf{Y}_2 \mathbf{E}_x(2) + \dots + \mathbf{Y}_p \mathbf{E}_x(p), \quad (5.10)$$

onde \mathbf{S} é a matriz que conterà uma representação holográfica de todos os vetores de saída.

Para recuperar o vetor \mathbf{y}_i , dado o vetor \mathbf{x}_i , utiliza-se a matriz randômica associada a $\mathbf{E}_x(i)$ e, através da aplicação da expressão (5.7) em (5.10), obtém-se:

$$\mathbf{Y}_i = \mathbf{S} \mathbf{E}_x^T(i). \quad (5.11)$$

O vetor de saída desejado, \mathbf{y}_i , é derivado diretamente dos elementos da diagonal principal da matriz \mathbf{Y}_i , como demonstraremos na próxima seção.

Assim, um conjunto de dados (caracterizados pelos pares de vetores de entrada e saída) é representado por uma única matriz (\mathbf{S}), com o auxílio das relações (5.10) e (5.11), compondo o que denominamos memória associativa holográfica (MAH).

5.5 Princípio de Operação da Memória Associativa Holográfica

A seguir, descreveremos com maiores detalhes o princípio de operação da memória associativa holográfica para o armazenamento e a recuperação de um par de vetores. Os conceitos matemáticos empregados são muito simples, o que só vem a salientar a simplicidade do modelo. Voltamos a realçar que o ponto chave da utilização deste tipo de memória consiste na utilização de duas representações simultâneas para cada dado: um vetor binário e uma memória aleatória associada:

Seja um par de vetores \mathbf{a} e \mathbf{b} , assim definidos:

$$\mathbf{a} = [a_1, a_2, \dots, a_n]$$

$$\mathbf{b} = [b_1, b_2, \dots, b_n],$$

onde os elementos a_i e b_i assumem apenas dois valores $[+1, -1]$ (representação binária na base de Rademacher [56]).

A partir destes vetores, definimos duas matrizes \mathbf{A} , \mathbf{B} , onde:

$$\mathbf{A} = [\mathbf{a}_{ij}] = \begin{cases} a_i, & i = j \\ 0, & i \neq j \end{cases}, \quad i, j = 1, 2, \dots, n \quad (5.12)$$

$$\mathbf{B} = [\mathbf{b}_{ij}] = \begin{cases} b_i, & i = j \\ 0, & i \neq j \end{cases}, \quad i, j = 1, 2, \dots, n \quad (5.13)$$

Além disso, associamos aos vetores \mathbf{a} e \mathbf{b} , respectivamente, duas matrizes \mathbf{E}_a e \mathbf{E}_b . Essas matrizes são constituídas de vetores randômicos (normalizados) com uma distribuição gaussiana de média zero.

De posse destas matrizes, definimos então a matriz \mathbf{S} , dada por:

$$\mathbf{S} = \mathbf{A}\mathbf{E}_b + \mathbf{B}\mathbf{E}_a. \quad (5.14)$$

Assim, dada a matriz randômica associada a um dos vetores, extrai-se o outro vetor através de uma das seguintes expressões:

$$\mathbf{A} \cong \mathbf{S}\mathbf{E}_b^T \quad (5.15)$$

$$\mathbf{B} \cong \mathbf{S}\mathbf{E}_a^T. \quad (5.16)$$

A fim de demonstrar a validade das expressões acima, substituímos a equação (5.16) em (5.13), obtendo:

$$\mathbf{A} \cong \mathbf{S}\mathbf{E}_b^T = \mathbf{A}\mathbf{E}_b\mathbf{E}_b^T + \mathbf{B}\mathbf{E}_a\mathbf{E}_b^T. \quad (5.17)$$

O produto de \mathbf{E}_b por sua transposta leva a:

$$\mathbf{E}_b\mathbf{E}_b^T \cong \mathbf{I}. \quad (5.18)$$

Da mesma forma, o produto de \mathbf{E}_a por \mathbf{E}_b^T leva a:

$$\mathbf{E}_a\mathbf{E}_b^T \cong 0. \quad (5.19)$$

Agora, aplicando (5.16) e (5.17) em (5.15), obtém-se:

$$\mathbf{A} \cong \mathbf{S}\mathbf{E}_b^T = \mathbf{A}\mathbf{I} + \boldsymbol{\delta}, \quad (5.20)$$

onde $\boldsymbol{\delta}$ é uma matriz de erro aleatório sobreposto ao sinal recuperado.

A demonstração de (5.16) pode ser obtida de forma análoga.

Para recuperar o vetor original de entrada \mathbf{a} (ou \mathbf{b}) a partir da matriz \mathbf{A} (ou \mathbf{B}), basta aplicar o seguinte procedimento:

$$a_i = \begin{cases} 1 & , a_{ii} \geq \mu \\ 0 & , -\mu \leq a_{ii} < \mu \\ -1 & , a_{ii} < -\mu \end{cases} \quad , i = 1, 2, \dots, n, \quad (5.21)$$

onde μ é um valor ($0 \leq \mu < 1$) que define um limiar de erro de recuperação de dados. A condição de erro é sinalizada por um elemento zero no vetor de saída.

A generalização deste procedimento, para um número qualquer de vetores, torna-se uma simples extensão do que foi demonstrado.

5.6 Robustez e Capacidade de Armazenamento da MAH

Considerando que a MAH representa os dados de forma holográfica, a informação é preservada mesmo quando existe uma perda de parte significativa da memória ou quando é corrompida por ruído. Armazenando um conjunto de dados composto por um único quadro, realizou-se os seguintes testes para avaliar a robustez da MAH:

- adição de ruído aleatório a todos os elementos da memória;
- destruição da informação em elementos selecionados de forma aleatória.

A Tabela 19 ilustra o desempenho da MAH, para diversas dimensões de memória, sendo apresentados os limites de razão sinal-ruído (SNR) e os correspondentes limites de destruição do conteúdo da memória, sem que haja perdas de informações relevantes. Observa-se, desses resultados, que os dados podem ser fortemente corrompidos sem o comprometimento da informação armazenada na memória.

TABELA 19
INDICADORES DE ROBUSTEZ DA MAH OBTIDOS EXPERIMENTALMENTE

Dimensões da MAH	Limite da SNR sem perda de informação	Limite de destruição sem perda de informação
50×50	-10,5 dB	85%
100×100	-12,5 dB	90%
150×150	-14,5 dB	94%
200×200	-16,0 dB	97%

É bom lembrar que a grande robustez da MAH tem um preço, que é a sua baixa capacidade de armazenamento (se comparada com memórias convencionais), pois, para cada dado armazenado, os demais dados podem ser considerados um “ruído” que tende a perturbar no processo de recuperação. Assim, se adicionarmos cada vez mais dados, eles começarão a ser perdidos. Podemos calcular a capacidade de armazenamento de uma MAH através da fórmula a seguir, que está deduzida no Apêndice 2:

$$p \cong \frac{n^2 / m}{\ln \left[\frac{2}{\alpha^2 \pi} \right] - 1}. \quad (5.22)$$

Onde:

m = dimensão do vetor de armazenamento;

n = dimensão da MAH ($n \times n$);

α = probabilidade admissível de que ocorra um de erro de recuperação;

p = número máximo de pares armazenados.

Podemos observar que a capacidade de armazenamento da MAH é diretamente proporcional à dimensão da matriz S e inversamente proporcional à dimensão do vetor de saída y_k . Por exemplo, uma MAH composta por uma matriz de dimensão 100×100 , que armazene vetores de dimensão 10, considerando-se uma probabilidade de extração correta superior a 99,9% ($\alpha = 0,001$), pode armazenar no máximo 80 pares de vetores. Se considerarmos que cada elemento da matriz é representado por 16 bits, teremos um total de 160000 bits utilizados para armazenar apenas 80 pares de informações com 10 bits por par (1600 bits no total). Assim, a MAH despende em média 100 vezes mais espaço para armazenar uma informação do que uma memória convencional. É bom lembrar que esta característica é inerente à maior parte das memórias associativas, sendo amplamente compensada pela robustez da MAH.

5.7 Aplicações da Memória Associativa Holográfica

A MAH pode ser utilizada em uma ampla gama de aplicações. A seguir, descreveremos de forma sucinta algumas dessas aplicações. Uma visão bastante completa e detalhada desse tipo de aplicação pode ser encontrada no trabalho de Plate [57].

5.7.1 Armazenamento de Seqüências

A MAH pode armazenar seqüências de sinais como, por exemplo, a seqüência a, b, c, d , que pode ser representada pela seguinte expressão:

$$S = \mathbf{AE}(1) + \mathbf{BE}(2) + \mathbf{CE}(3) + \mathbf{DE}(4). \quad (5.23)$$

5.7.2 Armazenamento de Conjunto de Seqüências

Conjuntos de seqüências, como, por exemplo $[a,b,c,d]$, $[f,g,a]$ e $[b,c,h]$, podem ser armazenados por:

$$\begin{aligned} \mathbf{S} = & (\mathbf{AE}_1(1) + \mathbf{BE}_1(2) + \mathbf{CE}_1(3) + \mathbf{DE}_1(4))\mathbf{E}_2(1) + \\ & (\mathbf{FE}_1(1) + \mathbf{GE}_1(2) + \mathbf{AE}_1(3))\mathbf{E}_2(2) + \\ & (\mathbf{BE}_1(1) + \mathbf{CE}_1(2) + \mathbf{HE}_1(3))\mathbf{E}_2(3). \end{aligned} \quad (5.24)$$

5.7.3 Representação de Variáveis

Em muito casos, desejamos armazenar relações do tipo:

$$x = a, y = b, z = c,$$

onde x, y, z são variáveis internas ao sistema e a, b, c representam valores para um dado instante.

Para esta representação, basta associar uma matriz randômica para cada variável. Assim,

$$\mathbf{S} = \mathbf{AE}(x) + \mathbf{BE}(y) + \mathbf{CE}(z). \quad (5.25)$$

5.7.4 Representação de Predicados

Para representar predicados do tipo: “*João come peixe*”, formamos uma tripla:

Tipo_de_ação (sujeito, predicado).

Se, para cada tipo de ação, associamos uma matriz randômica, teremos:

$$\mathbf{S} = \mathbf{M}_{Joao} \mathbf{E}(sujeito) \mathbf{E}(comer) + \mathbf{M}_{Peixe} \mathbf{E}(predicado) \mathbf{E}(Joao), \quad (5.26)$$

onde \mathbf{M}_{Joao} é uma matriz associada ao vetor binário que representa o símbolo “*João*”, e $\mathbf{E}(sujeito)$ é uma matriz randômica associada ao símbolo “*sujeito*”, sendo a mesma composição repetida para os demais símbolos.

5.7.5 Representação de Quadros

Informações definidas por quadros (*frames*) podem ser representadas como sentenças lógicas da seguinte forma:

quadro 1: (atributo 1, atributo 2, atributo 3,...).

Exemplo:

quadro *frutas*: (nome, cor, gosto)=

{(maçã, vermelha, doce), (limão, verde, ácido)}.

Para armazenar este quadro na MAH, temos:

$$\begin{aligned} \mathbf{S} = & \mathbf{M}_{maca} \mathbf{E}(gosto) \mathbf{E}(doce) + \mathbf{M}_{maca} \mathbf{E}(cor) \mathbf{E}(vermelha) + \\ & \mathbf{M}_{doce} \mathbf{E}(gosto) \mathbf{E}(maca) + \mathbf{M}_{vermelha} \mathbf{E}(cor) \mathbf{E}(maca) + \\ & \mathbf{M}_{limao} \mathbf{E}(gosto) \mathbf{E}(acido) + \mathbf{M}_{limao} \mathbf{E}(cor) \mathbf{E}(verde) + \\ & \mathbf{M}_{acido} \mathbf{E}(gosto) \mathbf{E}(limao) + \mathbf{M}_{verde} \mathbf{E}(cor) \mathbf{E}(limao). \end{aligned} \quad (5.27)$$

Para recuperar uma dada informação, basta utilizar as matrizes randômicas associadas às informações relativas aos dados armazenados. Dessa forma, a questão “Cor da maçã?” equivale a:

$$\mathbf{M}_{vermelha} = \mathbf{S} \mathbf{E}^T(maca) \mathbf{E}^T(cor). \quad (5.28)$$

5.8 Resultados Experimentais

Como uma das aplicações mais complexas da MAH consiste no armazenamento de quadros, essa aplicação foi utilizada em alguns casos reais que empregavam esse tipo de representação. A MAH foi utilizada para a representação de quadros que continham símbolos associados a vetores binários de dimensão 10. Foram utilizadas matrizes randômicas com dimensão 100×100, cujos elementos foram obtidos de um processo aleatório com distribuição normal (média zero e desvio padrão⁴ 0,1). A cada símbolo

⁴ Com um desvio de 0,1, uma matriz aleatória de 100x100 elementos, ao ser multiplicada por sua transposta, gera uma matriz muito próxima da matriz identidade.

associou-se uma dessas matrizes randômicas. A memória foi utilizada então para o armazenamento de um conjunto de quadros, como por exemplo:

*a cor do pino é azul,
a cor da bola é preta,
o tamanho do pino é pequeno.*

A recuperação das informações é realizada com questões (Q) e respostas (R) do tipo:

*Q: cor pino? R: azul
Q: cor pino amarelo? R: não
Q: cor pino azul? R: sim*

A memória implementada foi utilizada para armazenar até 60 conjuntos de dados sem que nenhuma perda de informação fosse verificada.

Em casos de informações conflitantes (por exemplo, para as sentenças do tipo: “*a cor do pino é azul*”, “*a cor do pino é vermelha*”, perguntamos qual é a cor do pino?), a resposta tende a ser um valor não-válido e, com menor frequência, tende a uma das duas respostas.

Se for aplicado um decaimento exponencial (fator de esquecimento) às informações existentes a cada vez que uma nova informação é armazenada, as informações menos recentes irão sendo “esquecidas”, o que torna a aplicação da MAH interessante para tratar informações em ambientes dinâmicos.

5.9 Plausibilidade Biológica

Observamos que a estrutura proposta pode ser implementada, utilizando RNAs nas quais os pesos dos neurônios definem as “matrizes” randômicas, e os estados de saída de um grande número de neurônios definem a “matriz” que representa a memória holográfica.

Além das diversas possíveis aplicações práticas para a MAH, pode-se extrapolar a concepção do modelo holográfico proposto. Consideremos então algumas características da MAH:

- restauração associativa de dados;

- representação distribuída (holográfica) das informações;
- robustez à perda de elementos de armazenamento;
- possibilidade de esquecimento;
- capacidade de tratamento de informações ambíguas;
- capacidade de processamento simbólico;
- possibilidade de implementação através de estruturas neurais.

Além disso, quando observamos os dados contidos nos elementos de memória (que são padrões aleatórios), não identificamos diretamente a informação ali armazenada. Essa só poderá ser recuperada com o auxílio da matriz randômica utilizada na fase de armazenamento.

A comparação de todas estas características com aquelas conjecturadas para a memória humana nos leva a pensar que este tipo de processamento holográfico possa lançar alguma luz sobre as obscuras questões de operação de nossa memória e de nosso cérebro como um todo.

5.10 Conclusões

A estrutura de memória associativa holográfica aqui discutida representa uma evolução em relação às estruturas existentes. Ela não necessita de nenhum processo de “limpeza” dos dados de saída, sendo assim uma evolução da HRR.

Os conceitos básicos da MAH são amplamente conhecidos, mas, para aplicá-los, tivemos de assumir que os símbolos manipulados pela memória são representados por duas formas distintas: um vetor binário e uma matriz randômica associada.

Para uma aplicação prática, é necessário utilizar matrizes de grandes dimensões. Dessa forma, a restauração dos dados demandará um considerável esforço computacional. Por outro lado, esse processamento pode ser realizado em um único passo, por uma máquina com processamento paralelo de grande capacidade.

O estado da arte atual das redes neurais ainda não permite uma implementação direta da MAH, sendo para tal necessário mais algum esforço de pesquisa. Observando que as matrizes aleatórias utilizadas na MAH podem ser implementadas por conjuntos de neurônios com pesos gerados randomicamente, visualizamos a possibilidade de

implementação de uma estrutura de processamento baseada em RNAs multicamadas na qual o treinamento das redes será diferente do treinamento gerado com algoritmos do tipo *backpropagation*.

Visualizamos também a possibilidade de aplicar a MAH como um “barramento de dados” que permitiria o transporte de informações entre estruturas de processamento neurais, transformando informações localizadas em informações distribuídas e *vice-versa*.

Capítulo 6

Conclusões

Podemos afirmar que a meta primordial deste trabalho de pesquisa foi contribuir para a criação de sistemas artificiais efetivamente “inteligentes”. Estamos cientes de que essa é uma meta ambiciosa e talvez por muito tempo ainda inatingível plenamente, a despeito dos milhares de pesquisadores que atualmente a perseguem. Apesar disso, temos plena convicção de que é possível desenvolver novos sistemas de processamento que permitirão a criação de “Inteligências Artificiais” com capacidade de aprendizado, raciocínio e criatividade semelhantes a dos seres humanos.

Não teríamos espaço para discutir aqui todas as grandes potencialidades e perigos que os sistemas artificiais inteligentes poderão trazer para a humanidade, mas uma simples analogia pode nos dar idéia da importância desse tipo de revolução tecnológica. A revolução industrial foi fruto do desenvolvimento de uma série de máquinas que liberaram o homem de inúmeros esforços físicos, gerando uma série de conseqüências que vão da produção em massa até a destruição dos recursos naturais do planeta. A revolução que está sendo feita nas áreas de informática e telecomunicação está transformando o mundo em uma aldeia global. Agora, as novas gerações de máquinas estão liberando não somente os músculos do homem, mas também seu cérebro. Por outro lado, a crescente evolução da tecnologia e o aumento da complexidade dos sistemas tende a demandar cientistas, técnicos e engenheiros cada vez mais especializados para dar continuidade ao processo. Por outro lado, essa extrema especialização demanda um esforço de treinamento cada vez maior e torna os profissionais pouco flexíveis. Neste cenário, o surgimento de sistemas artificiais inteligentes que possam substituir com vantagens técnicos humanos nas tarefas de operação e manutenção de sistemas automatizados será apenas um passo para um modelo no qual nossas máquinas projetem as novas gerações de máquinas, até um nível em que os seres humanos não tenham mais capacidade de entender a própria tecnologia que

geraram. Assim, como o homem do fim da idade média não podia imaginar o novo mundo que seria criado pela revolução industrial, nós também ainda não visualizamos totalmente o novo mundo que está sendo gerado pela revolução da informática, mas independentemente dos benefícios e dos riscos inerentes ao desenvolvimento de “inteligências artificiais”, com capacidades intelectuais iguais ou superiores às dos seres humanos, temos a certeza de que, se esses sistemas “inteligentes” puderem ser criados, eles de fato o serão para, no mínimo, mostrarmos que somos capazes de concebê-los.

6.1 Resultados Obtidos

O desenvolvimento da presente tese veio reforçar a visão de que as técnicas tradicionais de IA representam um “beco sem saída” para a evolução de sistemas “inteligentes”. Mais do que nunca, acreditamos que esses sistemas só poderão ser obtidos através de novas estruturas de processamento definidas com base em novos paradigmas de processamento simbólico e conexionista.

Neste contexto, tivemos a grande oportunidade de “romper com um paradigma” que é tão básico que chega a se insinuar, não como um paradigma, mas sim como a verdade absoluta. Desde cedo os nossos técnicos aprendem que tudo pode ser dividido em “caixas pretas”, descritas por um modelo de “entrada – processamento – saída”. Essa divisão nos parece tão natural que é muito difícil se afastar desse modelo, que é amplamente explorado em todos os sistemas de processamento criados e programados por seres humanos. Em sistemas autônomos, que aprendem a resolver problemas, uma hierarquização, baseada em “caixas pretas” conectadas por sinais de entrada e saída, não pode ser facilmente implementada, pois seria muito complexo treinar simultaneamente uma série de “caixas pretas” interligadas. Mesmo um algoritmo como o *backpropagation*, que treina com sucesso (mas lentamente) redes neurais com três camadas, será ineficiente se estivermos lidando com redes de dez ou vinte camadas. Assim, normalmente, os agentes autônomos, que recebem informações por meio de um processo de treinamento supervisionado, são modelados por uma única função que define completamente cada ponto no espaço de saída a partir do ponto observado no espaço de entrada. Dessa forma, o modelo proposto, além de definir uma nova estrutura de treinamento para agentes cognitivos, permite a resolução de problemas complexos para os quais o modelo

tradicional é praticamente inviável.

Apesar da parte experimental deste trabalho ter lidado apenas com problemas de complexidade média, os conceitos utilizados no tratamento dos problemas são genéricos e, dessa forma, não seria muito difícil, a partir do agente que resolve o problema do “Jogo da Velha”, criar um agente que, por exemplo, jogue Xadrez, sendo as maiores dificuldades mais relacionadas ao volume de trabalho para a definição do processo de treinamento do que para a definição da estrutura de controle do agente. Infelizmente, o volume de trabalho exigido na implementação de um agente que jogue Xadrez é muito elevado para ser realizado no contexto desta tese, mas seria um interessante trabalho futuro para validar o novo modelo aqui apresentado.

Os outros resultados apresentados são as escalas para medição da “inteligência” de sistemas artificiais e as memórias associativas holográficas. A criação de escalas que permitam a medição de parâmetros relacionados com o “grau de inteligência” de um sistema artificial deverá ser um marco fundamental para o efetivo desenvolvimento desse tipo de sistema. Após o desenvolvimento da presente tese, ficou muito claro que estudar e desenvolver sistemas inteligentes sem o uso de escalas para medição de inteligência equivale a tentar estudar mineralogia sem ter uma escala para medição de dureza de rochas ou estudar terremotos sem ter uma escala para medição de sua intensidade.

As duas escalas propostas nesta tese, apesar de rudimentares, demonstraram sua viabilidade para a avaliação de agentes cognitivos naturais ou artificiais. O desenvolvimento de escalas completas e amplamente aceitas extrapola o âmbito desta tese, mas acreditamos que as mesmas são tão necessárias para o desenvolvimento e a avaliação de novos modelos que sua efetivação é praticamente inevitável.

A Memória Associativa Holográfica tem bases matemáticas muito simples, mas representa uma grande evolução em relação aos modelos de memórias associativas encontrados em literatura e também pode ser implementada por meio de estruturas neurais.

A MAH opera de modo “próximo” a como pensamos operar a memória humana, e acreditamos que o modelo proposto talvez possa servir como base para o tipo de memória que um sistema efetivamente inteligente deva possuir.

É interessante observar que a MAH armazena informações, distribuídas de forma holográfica, que podem ser recuperadas diretamente sem a necessidade de um processo de busca que é básico a qualquer sistema tradicional de processamento.

Idealizando alguns tipos de implementações da MAH baseadas em RNAs, observamos com satisfação que as matrizes aleatórias utilizadas no modelo podem ser facilmente obtidas através de neurônios que tenham seus pesos gerados segundo uma distribuição aleatória que nos parece bastante natural e nos leva a refletir sobre a plausibilidade biológica do modelo proposto.

6.2 Considerações Finais

Através de uma análise histórica do desenvolvimento científico das diversas áreas do conhecimento humano, percebemos que sempre existem vários “caminhos” a serem explorados, mas, em alguns casos, para que haja um efetivo avanço em uma determinada área outros “caminhos” devem ser prospectados.

Assim, acreditamos que as “Inteligências Artificiais” que buscamos deverão ser baseadas em estruturas de processamento totalmente diversas dos modelos atualmente utilizados, mas que possivelmente terão algumas das seguintes características:

- o sistema deverá aprender em um processo contínuo e interativo, dentro de um ambiente real ou virtual;
- o processo de aprendizagem deverá ser bastante similar ao dos seres humanos;
- o sistema terá algum tipo de interpretação concreta para os dados e símbolos que manipula, relacionada com o mundo no qual está inserido;
- dificilmente os programadores do sistema saberão ao certo quais informações estão sendo armazenadas ou manipuladas pelo sistema;
- os dados estarão representados de diversas formas, podendo estar armazenados em estruturas de memória associativas;
- as formas de processamento serão eminentemente paralelas e talvez holográficas.

Desta forma, na nossa visão, alguns “caminhos” deverão ser seguidos no desenvolvimento deste tipo de sistema como, por exemplo:

- desenvolvimento de escalas para medição de “inteligência”;
- rompimento dos paradigmas de processamento tradicionais e mais amplo desenvolvimento dos modelos conexionistas existentes;

- desenvolvimento de novos modelos de agentes cognitivos e de novas formas de treinamento;
- desenvolvimento de novas estruturas de memória e de processamento holográfico.

Assim, acreditamos que, através da presente tese, alguns passos importantes foram dados na direção que levará à criação de sistemas efetivamente “inteligentes”.

Ao concluir este trabalho, observamos que os resultados obtidos superaram em muito a nossa expectativa original, principalmente no tocante à definição de um novo paradigma de processamento, que amplia o tradicional modelo de “entrada – processamento – saída”. Sabemos que isto se deve ao fato de termos buscado subsídios não só nas áreas de engenharia e computação, mas também nas áreas humanas de psicologia, filosofia e lingüística, principalmente, no modelo de Semiótica de Charles Peirce e no modelo Construtivista de Jean Piaget. Essa busca gerou os resultados apresentados nesta tese e deu-nos maior convicção de que os sistemas efetivamente inteligentes representam uma possibilidade concreta que, para ser atingida, demandará novas visões que deverão ser principalmente holísticas e interdisciplinares, pois, no caso do cérebro humano e dos sistemas efetivamente “inteligentes”, o todo é certamente muito maior que a soma das partes.

Apêndice 1

Exemplos de Aplicação de Metaesquemas de Transição

A.1 Introdução

Neste apêndice, descreveremos, de forma sucinta, algumas aplicações possíveis de serem implementadas através de esquemas de transição:

- problema da máquina caça-níqueis;
- controle de um agente que “come” objetos;
- controle de um “elevador”;
- problema do “Jogo da Velha”.

Todos estes problemas são relativamente simples, mas bastante ilustrativos, sendo definida, para tratar cada um deles, uma estrutura hierárquica de metaesquemas que são processados por um mesmo conjunto de rotinas.

A seguir, cada um destes problemas será abordado em maiores detalhes.

A.2 Problema da Máquina Caça-níqueis

Neste problema, um agente encontra-se diante de uma série de máquinas caça-níqueis (cada uma com um botão de acionamento). Ao ativar uma máquina, o agente pode “ganhar” com uma certa probabilidade inerente a cada máquina. O agente deve, após algumas jogadas, optar preferencialmente pela máquina com maior retorno. Como as probabilidades podem ser alteradas, o agente não deve jogar em uma única máquina, sem deixar de priorizar as que dão maior retorno.

O agente possui apenas uma entrada e quatro saídas:

E_1 - Reforço positivo;

S_1 - Jogar na máquina 1;

S_2 - Jogar na máquina 2;

S_3 - Jogar na máquina 3;

S_4 - Jogar na máquina 4.

A Fig. 16 mostra a tela de um simulador que implementa este problema. Na tela, apresentamos alguns dados estatísticos bem como os estados das entradas e saídas do agente. Quatro botões servem para a realização de um treinamento não supervisionado, que consiste em apertar um único botão, algumas vezes, como mostra a Fig. 17.

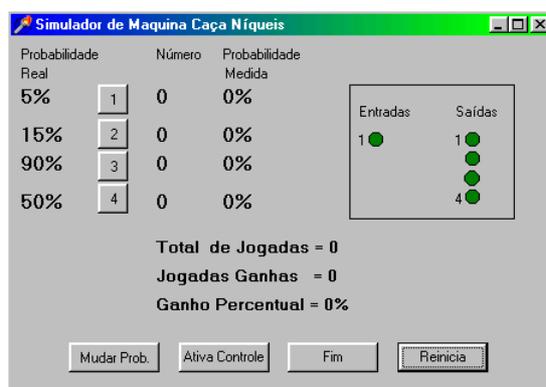


Fig. 16. Simulador do problema de caça-níqueis, mostrando estado inicial.

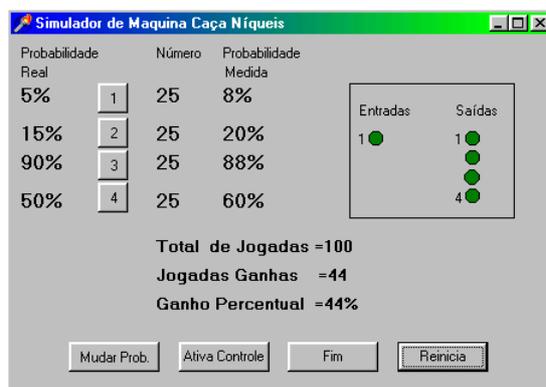


Fig. 17. Simulador após 100 jogadas definidas manualmente.

Podemos observar na Fig. 18 que, após realizar 279 jogadas, a maior parte dessas jogadas foram realizadas na máquina que apresenta maior probabilidade de vitória. Ainda assim, o agente continua jogando na máquina 4 e, com menor frequência, nas máquinas 1 e 2. Após 200 jogadas, as probabilidades das máquinas foram alteradas, no entanto, foram necessárias mais 600 jogadas para que o agente perceba as novas probabilidades. Isso

ocorre devido às possibilidades calculadas no metaesquema considerarem todas as jogadas já realizadas. Uma estratégia interessante para ambientes mais dinâmicos é a utilização de um fator de esquecimento, considerando apenas as possibilidades das últimas transições observadas.

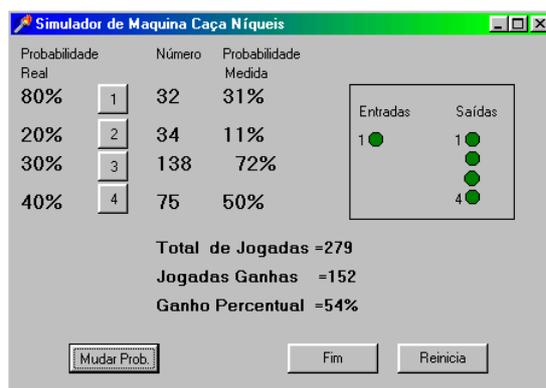


Fig. 18. Situação após 279 jogadas e efetivadas pelo agente. A probabilidade das máquinas foi modificada na jogada 200.

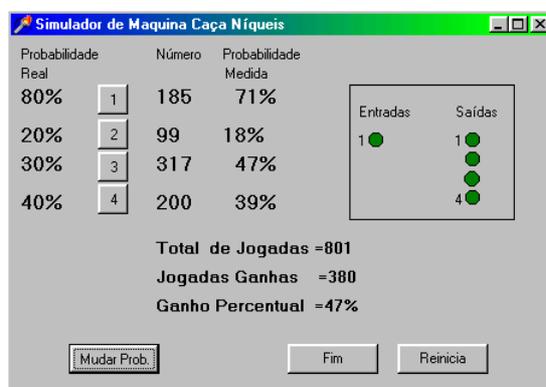


Fig. 19. Situação após 801 jogadas e efetivadas pelo agente.

O programa de controle definido para este problema é composto de um único MET, conforme mostra a Tabela 20. Observamos que o sinal de reforço gera um espaço de efeito unidimensional, sendo o objetivo de entrada obtido referente a um reforço igual a 1.

TABELA 20
PROGRAMA DE CONTROLE DO PROBLEMA DA MÁQUINA CAÇA -NÍQUEIS

% Controle da máquina de caça-níqueis
% Espaço de causa = S1, S2, S3, S4 = máquinas 1 a 4
% Espaço de efeito = E1 = vitória
% Saída de objetivos = S1, S2, S3, S4
% Entrada de objetivos = 1
(S1, S2, S3, S4 E1 S1, S2, S3, S4 1)
FIM

O MET irá mapear as transições mostradas na Tabela 21. Essas transições demonstram o efeito de uma saída sendo ativada a cada vez. Como, no processo de treinamento, nunca permitimos a ativação simultânea de mais de uma saída, o agente não aprende a ativá-las dessa forma. É fácil observar que o MET deverá optar entre quatro possíveis “caminhos” para atingir seu objetivo. Cada “caminho” irá participar de um sorteio. Aqueles de maior possibilidade terão mais chances de serem sorteados. Como os objetivos de saída são os próprios sinais de saída, eles são automaticamente convertidos em ações que equivalem a jogar em uma das quatro máquinas. Esse problema é bastante simples, mas ilustra alguns dos principais mecanismos de controle de um metaesquema de transição.

TABELA 21
TRANSIÇÕES OBSERVADAS PELO MET NAS MÁQUINAS CAÇA -NÍQUEIS

Espaço de Causa (máquinas 1 a 4)		Espaço de Efeito Inicial (vitória)		Possibilidade
Causa Inicial	Causa Final	Efeito Inicial	Efeito Final	
0, 0, 0, 0	1, 0, 0, 0	0	1	0,7999
0, 0, 0, 0	0, 1, 0, 0	0	1	0,2089
0, 0, 0, 0	0, 0, 1, 0	0	1	0,4401
0, 0, 0, 0	0, 0, 0, 1	0	1	0,4216

A.3 Controle de um Agente que “Come” Objetos

Neste problema, um agente encontra-se diante de uma máquina com uma tela e dois botões. Um botão mostra um novo objeto e outro “come” um objeto. A tela mostra cinco tipos de objetos (quadrado vermelho, quadrado verde, bola vermelha, bola verde e

“objeto nulo”). Se o agente comer um quadrado vermelho, ele recebe um estímulo de prazer e, se comer qualquer outro objeto, recebe um estímulo de dor.

O agente possui seis entradas e duas saídas:

E_1 - Quadrado;

E_2 - Bola;

E_3 - Objeto Vermelho;

E_4 - Objeto Verde;

E_5 - Reforço negativo;

E_6 - Reforço positivo.

S_1 - Comer objeto;

S_2 - Próximo objeto.

A Fig. 20 mostra a tela de um simulador que implementa este problema. Nessa tela, apresentamos os objetos, os dois botões, os sinais de reforço e, também, os estados das entradas e saídas do agente. Os dois botões servem para a realização de um treinamento não supervisionado.

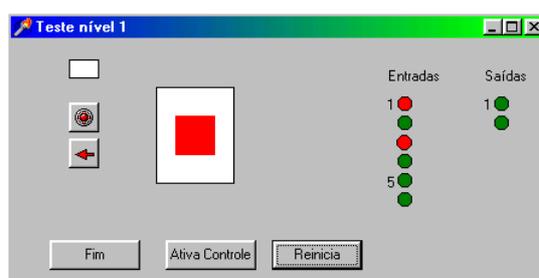


Fig. 20. Simulador para o problema de “comer objetos”.

O programa de controle definido para este problema é composto de dois METs, conforme mostra a Tabela 22. O primeiro MET irá relacionar os estímulos de dor e prazer com os objetos apresentados e o segundo MET irá relacionar os objetos apresentados com a ativação dos botões.

TABELA 22
PROGRAMA DE CONTROLE DO PROBLEMA DA MÁQUINA CAÇA -NÍQUEIS

% Espaço de causa = E1, E2, E3, E4 = quadrado, bola, vermelho, verde
% Espaço de efeito = E5, E6 = dor, prazer
% Entrada de objetivos = 0, 1
% Entrada de objetivos negados = 1, 0
% Saída de objetivos = O1, O2, O3, O4
(E1, E2, E3, E4 E5, E6 O1, O2, O3, O4 0, 1 1, 0)
% Espaço de causa = S1, S2 = comer objeto, próximo objeto
% Espaço de efeito = E1, E2, E3, E4 = quadrado, bola, vermelho, verde
% Entrada de objetivos = O1, O2, O3, O4
% Saída de objetivos = S1, S2
(S1, S2 E1, E2, E3, E4 S1, S2 O1, O2, O3, O4)
FIM

O primeiro MET irá mapear as transições mostradas na Tabela 23. Nessas transições, observamos que a única transição que causa prazer é a que leva de um quadrado vermelho a um “objeto nulo”. Dessa forma, a saída objetivo desse MET será primeiro um quadrado vermelho e a seguir um “objeto nulo”. O segundo MET mapeia as funções dos botões, verificando que o primeiro botão leva sempre a um “objeto nulo” e que o segundo leva de um tipo de objeto para outro com uma possibilidade próxima a 25%. A definição de um objetivo negado é importante para bloquear um caminho que seria “comer” o objeto e a seguir pedir um novo objeto.

TABELA 23
TRANSIÇÕES OBSERVADAS PELO PRIMEIRO MET

Espaço de Causa (quadrado, bola, vermelho, verde)		Espaço de Efeito Inicial (dor, prazer)		Possibilidade
Causa Inicial	Causa Final	Efeito Inicial	Efeito Final	
0, 1, 1, 0	0, 0, 0, 0	0, 0	1, 0	1,0000
1, 0, 0, 1	0, 0, 0, 0	0, 0	1, 0	1,0000
1, 0, 1, 0	0, 0, 0, 0	0, 0	0, 1	1,0000
0, 1, 0, 1	0, 0, 0, 0	0, 0	1, 0	1,0000

TABELA 24
TRANSIÇÕES OBSERVADAS PELO SEGUNDO MET

Espaço de Causa (comer, próximo)		Espaço de Efeito Inicial (quadrado, bola, vermelho, verde)		Possibilidade
Causa inicial	Causa Final	Efeito Inicial	Efeito Final	
0, 0	1, 0	0, 1, 1, 0	0, 0, 0, 0	1,00
0, 0	1, 0	1, 0, 0, 1	0, 0, 0, 0	1,00
0, 0	1, 0	1, 0, 1, 0	0, 0, 0, 0	1,00
0, 0	1, 0	0, 1, 0, 1	0, 0, 0, 0	1,00
0, 0	0, 1	0, 0, 0, 0	0, 1, 0, 1	0,27
0, 0	0, 1	0, 1, 0, 1	0, 1, 1, 0	0,10
0, 0	0, 1	0, 1, 1, 0	1, 0, 0, 1	0,22
0, 0	0, 1	1, 0, 0, 1	0, 1, 0, 1	0,27
0, 0	0, 1	0, 1, 0, 1	1, 0, 0, 1	0,12
0, 0	0, 1	1, 0, 1, 0	1, 0, 0, 1	0,19
0, 0	0, 1	1, 0, 1, 0	0, 1, 0, 1	0,16
0, 0	0, 1	0, 1, 0, 1	1, 0, 1, 0	0,17
0, 0	0, 1	1, 0, 1, 0	0, 1, 1, 0	0,10
0, 0	0, 1	0, 0, 0, 0	1, 0, 0, 1	0,23
0, 0	0, 1	0, 0, 0, 0	0, 1, 1, 0	0,27
0, 0	0, 1	0, 0, 0, 0	1, 0, 1, 0	0,24
0, 0	0, 1	0, 1, 1, 0	1, 0, 1, 0	0,16
0, 0	0, 1	0, 1, 1, 0	0, 1, 0, 1	0,16
0, 0	0, 1	1, 0, 0, 1	1, 0, 1, 0	0,18
0, 0	0, 1	1, 0, 0, 1	0, 1, 1, 0	0,15

A.4 Controle de um “Elevador”

Neste problema, um agente encontra-se em um “elevador” que o levará a três andares. Em cada um desses andares, dois tipos de objetos (bolas vermelhas e azuis) são arremessados sobre o agente. Se uma bola vermelha atinge o agente, ele sente “dor”, se uma bola azul o atinge, ele sente prazer. O agente possui dois botões que fazem o elevador subir ou descer um andar.

O agente possui oito entradas e duas saídas:

E_1 - Agente no primeiro andar;

E_2 - Agente no segundo andar;

E_3 - Agente no terceiro andar;

E_4 - Objeto no primeiro andar (0=sem objeto, 1=objeto azul, 2=objeto vermelho);

E_5 - Objeto no segundo andar (0=sem objeto, 1=objeto azul, 2=objeto vermelho);

E_6 - Objeto no terceiro andar (0=sem objeto, 1=objeto azul, 2=objeto vermelho);

E_7 - Reforço negativo;

E_8 - Reforço positivo;

S_1 - Sobe;

S_2 - Desce.

A Fig. 21 mostra a tela de um simulador que implementa este problema. Na tela, apresentamos os três andares, o agente, os objetos, os dois botões de controle do elevador e também os estados das entradas e saídas do agente. Os dois botões servem para a realização de um treinamento não supervisionado.

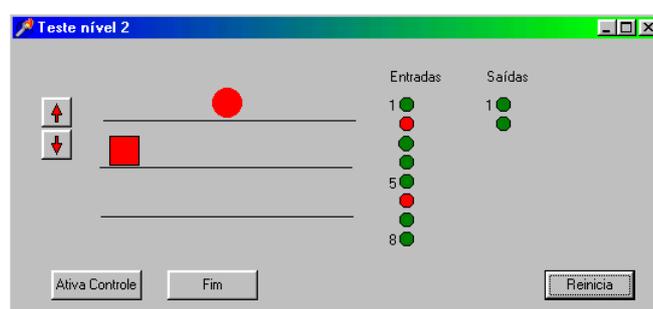


Fig. 21. Simulador para o controle do “elevador”.

O programa de controle definido para este problema é composto de dois METs, conforme mostra a Tabela 25. O primeiro MET irá relacionar os estímulos de dor e prazer com a posição do agente e a posição dos objetos, o segundo MET irá relacionar a posição do agente com os botões de controle do elevador.

TABELA 25
PROGRAMA DE CONTROLE DO AGENTE

```
% Controle do elevador
% Espaço de causa = E1, E2, E3, E4, E5, E6 = posição do agente, objeto
% Espaço de efeito = E7, E8 = dor, prazer
% Entrada de objetivos = 0, 1
% Entrada de objetivos negados = 1, 0
% Saída de objetivos = O1, O2, O3
(E1, E2, E3, E4, E5, E6 | E7, E8 | O1, O2, O3 | 0, 1 | 1, 0)
% Espaço de causa = S1, S2 = sobe, desce
% Espaço de efeito = E1, E2, E3 = posição do agente
% Entrada de objetivos = O1, O2, O3
% Saída de objetivos = S1, S2
(S1, S2 | E1, E2, E3 | S1, S2 | O1, O2, O3)
FIM
```

O primeiro MET irá mapear as transições mostradas na Tabela 26, nas quais observamos que a única transição que causa prazer é a que leva de um quadrado vermelho a um “objeto nulo”. Dessa forma, a saída objetivo desse MET será primeiro um quadrado vermelho e a seguir um “objeto nulo”. O segundo MET mapeia as funções dos botões, verificando que o primeiro botão leva sempre a um “objeto nulo” e que o segundo leva de um tipo de objeto para outro com uma possibilidade próxima a 25%. A definição de um objetivo negado é importante para bloquear um caminho que seria “pegar” o objeto e a seguir pedir um novo objeto.

TABELA 26
TRANSIÇÕES OBSERVADAS PELO PRIMEIRO MET

Espaço de Causa (E1, E2, E3, E4, E5, E6)		Espaço de Efeito Inicial (dor, prazer)		Possibilidade
Causa Inicial	Causa Final	Efeito Inicial	Efeito Final	
1, 0, 0, 2, 0, 0	1, 0, 0, 2, 0, 0	0, 0	0, 1	1,00
1, 0, 0, 2, 0, 0	1, 0, 0, 0, 0, 0	0, 1	0, 0	1,00
1, 0, 0, 1, 0, 0	1, 0, 0, 1, 0, 0	0, 0	1, 0	1,00
1, 0, 0, 1, 0, 0	1, 0, 0, 0, 0, 0	1, 0	0, 0	1,00
0, 1, 0, 0, 1, 0	0, 1, 0, 0, 1, 0	0, 0	1, 0	1,00
0, 1, 0, 0, 1, 0	0, 1, 0, 0, 0, 0	1, 0	0, 0	1,00
0, 1, 0, 0, 2, 0	0, 1, 0, 0, 2, 0	0, 0	0, 1	1,00
0, 1, 0, 0, 2, 0	0, 1, 0, 0, 0, 0	0, 1	0, 0	1,00
0, 0, 1, 0, 0, 2	0, 0, 1, 0, 0, 2	0, 0	0, 1	1,00
0, 0, 1, 0, 0, 2	0, 0, 1, 0, 0, 0	0, 1	0, 0	1,00
0, 0, 1, 0, 0, 1	0, 0, 1, 0, 0, 1	0, 0	1, 0	1,00
0, 0, 1, 0, 0, 1	0, 0, 1, 0, 0, 0	1, 0	0, 0	1,00

TABELA 27
TRANSIÇÕES OBSERVADAS PELO SEGUNDO MET

Espaço de Causa (sobe, desce)		Espaço de Efeito Inicial (E1, E2, E3)		Possibilidade
Causa inicial	Causa final	Efeito inicial	Efeito Final	
0, 0	1, 0	1, 0, 0	0, 1, 0	1,00
0, 0	0, 1	0, 1, 0	1, 0, 0	1,00
0, 0	1, 0	0, 1, 0	0, 0, 1	1,00
0, 0	0, 1	0, 0, 1	0, 1, 0	1,00

A.5 Problema do “Jogo da Velha”

O “Jogo da Velha” é um jogo de tabuleiro muito simples que todos conhecem. O agente observa um tabuleiro com nove posições onde cada uma delas pode assumir três estados (vazio, peça X, peça O). O agente possui nove saídas que indicam em qual casa ele pretende jogar. Além disso, o agente recebe sinais que indicam se ele venceu, perdeu ou empatou um jogo. Dois outros sinais são utilizados para sincronizar a seqüência de jogo: o

primeiro sinal indica que o agente deve jogar (para saber quem começa o jogo e também sincronizar cada jogada do agente) e o segundo, indica que a partida está em andamento.

Desta forma, o agente possui quatorze entradas:

- E₁ - Estado da casa 1;
- E₂ - Estado da casa 2;
- E₃ - Estado da casa 3;
- E₄ - Estado da casa 4;
- E₅ - Estado da casa 5;
- E₆ - Estado da casa 6;
- E₇ - Estado da casa 7;
- E₈ - Estado da casa 8;
- E₉ - Estado da casa 9;
- E₁₀ - Indica que o agente deve jogar;
- E₁₁ - Perdeu o jogo;
- E₁₂ - Venceu o jogo;
- E₁₃ - Fim do jogo;
- E₁₄ - Indica que a partida está em andamento.

E possui também nove saídas:

- S₁ - Jogar na casa 1;
- S₂ - Jogar na casa 2;
- S₃ - Jogar na casa 3
- S₄ - Jogar na casa 4;
- S₅ - Jogar na casa 5;
- S₆ - Jogar na casa 6;
- S₇ - Jogar na casa 7;
- S₈ - Jogar na casa 8;
- S₉ - Jogar na casa 9.

A Fig. 22 mostra a tela de um simulador do “Jogo da Velha”. Na tela, são apresentados um tabuleiro e botões para jogar, como também os estados das entradas e saídas do agente. A Fig. 23 ilustra uma fase do processo de treinamento em que o agente é ensinado a jogar através de partidas muito simples.

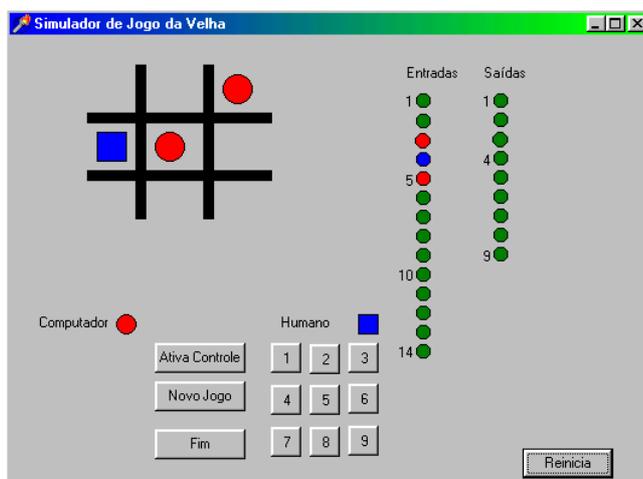


Fig. 22. Simulador do “Jogo da Velha”.

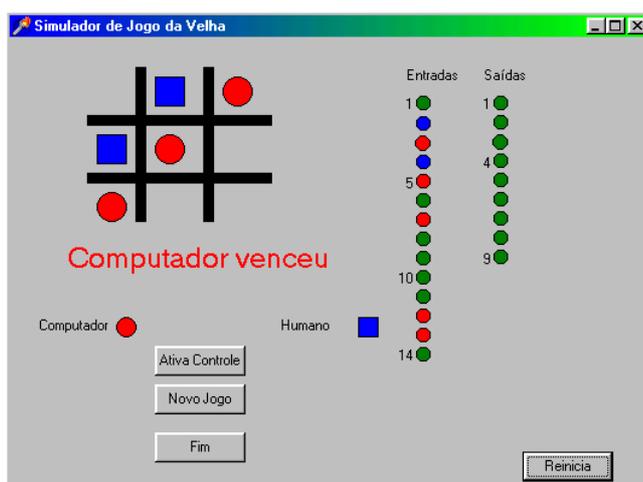


Fig. 23. Simulador do “Jogo da Velha” na fase de treinamento supervisionado.

O programa de controle definido para este problema é composto de 10 METs, conforme mostra a Tabela 28. O primeiro MET irá relacionar os sinais de vitória e derrota com as posições observadas no tabuleiro. Os demais METs controlam individualmente cada uma das saídas, aprendendo basicamente que não podem jogar em uma casa que já esteja ocupada.

TABELA 28
PROGRAMA DE CONTROLE DO AGENTE

```

% Programa do “Jogo da Velha”
% Espaço de causa = E1, E2, E3, E4, E5, E6, E7, E8,E9 = tabuleiro
% Espaço de efeito = E11 E12 = dor, prazer
% Entrada de objetivos = 0, 1
% Entrada de objetivos negados = 1, 0
% Saída de objetivos = O1, O2, O3, O4, O5, O6, O7, O8, O9
% O símbolo $CC indica que a transição inicial de causa não importa
( $CC, E1, E2, E3, E4, E5, E6, E7, E8,E9 | E11 E12 | O1, O2, O3, O4, O5, O6, O7, O8, O9 | 0 1 | 1 0)
% Nove MET's de controle de saída (x = 1,2,...,9)
% Espaço de causa = Sx, E10, E14 = saída x, jogar, partida em andamento
% Espaço de efeito = Ex
% Entrada de objetivos = Ox
% Saída de objetivos = Sx
(S1, E10, E14 | E1 | S1 | O1)
(S2, E10, E14 | E2 | S2 | O2)
(S3, E10, E14 | E3 | S3 | O3)
(S4, E10, E14 | E4 | S4 | O4)
(S5, E10, E14 | E5 | S5 | O5)
(S6, E10, E14 | E6 | S6 | O6)
(S7, E10, E14 | E7 | S7 | O7)
(S8, E10, E14 | E8 | S8 | O8)
(S9, E10, E14 | E9 | S9 | O9)
FIM

```

O primeiro MET irá mapear as transições mostradas na Tabela 29. Nessas transições, o agente irá observar quais são as configurações que geram vitória e quais os que geram derrota, como a causa inicial não importa, essa coluna não é registrada pelo agente. O objetivo “vencer o jogo” irá gerar um objetivo de saída equivalente a uma configuração vitoriosa e o objetivo “negado” dado por “perder o jogo” irá gerar jogadas de bloqueio. É importante observar que o estado “2” no tabuleiro, que indica a jogada do oponente, será tratado pelo agente como uma transição sem causa observável e, dessa forma, será tratada como uma causa aleatória.

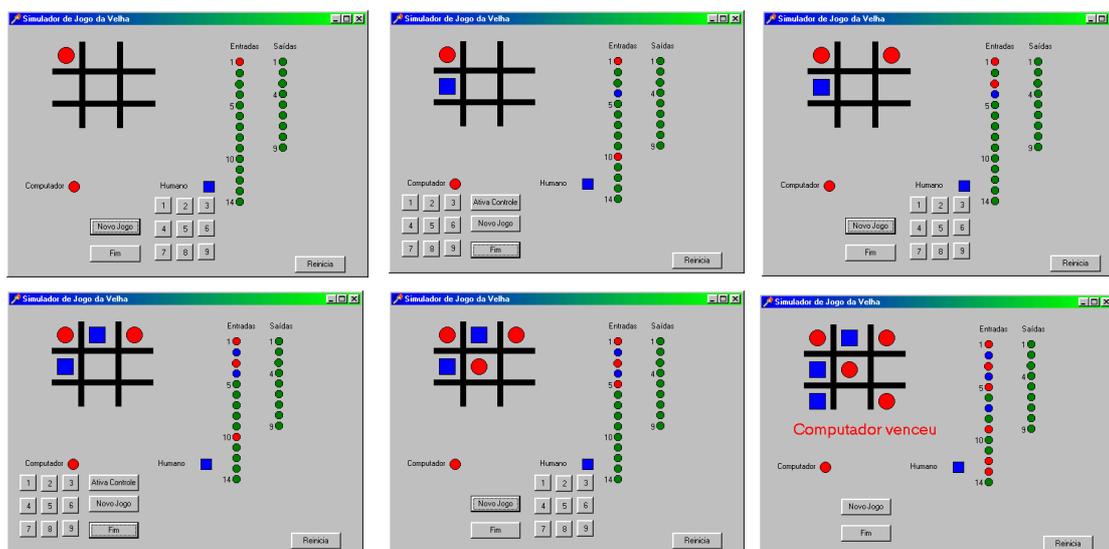


Fig. 24. Agente jogando contra um adversário humano.

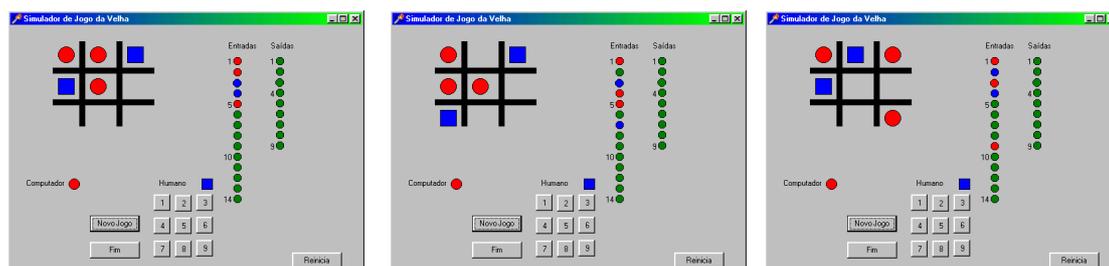


Fig. 25. Algumas das estratégias de jogo aprendidas pelo agente.

O segundo MET mapeia a E_1 como efeito da S_1 e de E_{10} e E_{14} , como mostra a Tabela 30. Esse esquema deve identificar que, quando for a vez do agente jogar ($E_{10} = 1$), uma ativação da saída leva a entrada a um valor igual a um. Ele identifica também que, quando não for a vez do agente ($E_{10} = 0$), a entrada E_1 pode ir para o valor 2 de forma aleatória (sem causa observada). Além disso, o MET verifica que a ativação de E_{14} leva E_1 para um valor nulo. Os demais metaesquemas terão tabelas de transição idênticas a do segundo MET.

TABELA 29
TRANSIÇÕES OBSERVADAS PELO PRIMEIRO MET

Espaço de Causa (nove casas do tabuleiro)		Espaço de Efeito Inicial (dor, prazer)		Possibilidade
Causa Inicial	Causa Final	Efeito Inicial	Efeito Final	
0, 0, 0, 0, 0, 0, 0, 0, 0	1, 2, 0, 1, 2, 0, 1, 0, 0	0, 0	0, 1	1,00
0, 0, 0, 0, 0, 0, 0, 0, 0	2, 1, 0, 2, 1, 0, 2, 0, 0	0, 0	1, 0	1,00
0, 0, 0, 0, 0, 0, 0, 0, 0	2, 1, 0, 2, 1, 0, 0, 1, 0	0, 0	0, 1	1,00
0, 0, 0, 0, 0, 0, 0, 0, 0	1, 2, 0, 1, 2, 0, 0, 2, 0	0, 0	1, 0	1,00
0, 0, 0, 0, 0, 0, 0, 0, 0	1, 1, 2, 2, 1, 0, 0, 2, 1	0, 0	0, 1	1,00
0, 0, 0, 0, 0, 0, 0, 0, 0	1, 2, 1, 2, 1, 2, 0, 0, 1	0, 0	0, 1	1,00
0, 0, 0, 0, 0, 0, 0, 0, 0	1, 1, 2, 2, 2, 1, 1, 2, 1	0, 0	0, 1	1,00
0, 0, 0, 0, 0, 0, 0, 0, 0	1, 1, 1, 2, 1, 0, 0, 2, 2	0, 0	0, 1	1,00
0, 0, 0, 0, 0, 0, 0, 0, 0	1, 1, 2, 2, 1, 0, 0, 1, 2	0, 0	0, 1	1,00
0, 0, 0, 0, 0, 0, 0, 0, 0	1, 2, 1, 2, 1, 0, 2, 0, 1	0, 0	0, 1	1,00
0, 0, 0, 0, 0, 0, 0, 0, 0	1, 1, 1, 2, 2, 0, 0, 0, 0	0, 0	0, 1	1,00
0, 0, 0, 0, 0, 0, 0, 0, 0	1, 1, 0, 2, 2, 2, 0, 1, 0	0, 0	1, 0	1,00
0, 0, 0, 0, 0, 0, 0, 0, 0	1, 2, 1, 1, 2, 2, 0, 2, 1	0, 0	1, 0	1,00
0, 0, 0, 0, 0, 0, 0, 0, 0	1, 0, 1, 2, 2, 2, 1, 0, 0	0, 0	1, 0	1,00
0, 0, 0, 0, 0, 0, 0, 0, 0	2, 1, 2, 0, 2, 0, 2, 1, 1	0, 0	1, 0	1,00
0, 0, 0, 0, 0, 0, 0, 0, 0	2, 1, 0, 0, 2, 0, 0, 1, 2	0, 0	1, 0	1,00
0, 0, 0, 0, 0, 0, 0, 0, 0	1, 2, 0, 1, 1, 1, 2, 0, 2	0, 0	0, 1	1,00
0, 0, 0, 0, 0, 0, 0, 0, 0	2, 1, 0, 1, 1, 0, 2, 2, 2	0, 0	1, 0	1,00
0, 0, 0, 0, 0, 0, 0, 0, 0	1, 2, 1, 2, 2, 1, 0, 0, 1	0, 0	0, 1	1,00
0, 0, 0, 0, 0, 0, 0, 0, 0	2, 1, 1, 2, 1, 0, 2, 0, 0	0, 0	1, 0	1,00
0, 0, 0, 0, 0, 0, 0, 0, 0	2, 1, 2, 0, 2, 0, 1, 1, 2	0, 0	1, 0	1,00
0, 0, 0, 0, 0, 0, 0, 0, 0	1, 1, 2, 0, 2, 1, 2, 0, 0	0, 0	1, 0	1,00
0, 0, 0, 0, 0, 0, 0, 0, 0	2, 1, 0, 2, 2, 1, 2, 0, 1	0, 0	1, 0	1,00
0, 0, 0, 0, 0, 0, 0, 0, 0	1, 0, 2, 1, 1, 1, 2, 0, 2	0, 0	0, 1	1,00
0, 0, 0, 0, 0, 0, 0, 0, 0	1, 2, 1, 2, 2, 1, 1, 1, 2	0, 0	0, 1	1,00
0, 0, 0, 0, 0, 0, 0, 0, 0	1, 2, 1, 2, 2, 1, 2, 1, 2	0, 0	0, 1	1,00
0, 0, 0, 0, 0, 0, 0, 0, 0	1, 0, 2, 2, 2, 0, 1, 1, 1	0, 0	0, 1	1,00
0, 0, 0, 0, 0, 0, 0, 0, 0	1, 1, 2, 2, 2, 2, 1, 0, 1	0, 0	1, 0	1,00
0, 0, 0, 0, 0, 0, 0, 0, 0	2, 2, 1, 1, 2, 2, 1, 2, 1	0, 0	1, 1	1,00
0, 0, 0, 0, 0, 0, 0, 0, 0	1, 0, 2, 1, 1, 2, 2, 0, 1	0, 0	0, 1	1,00
0, 0, 0, 0, 0, 0, 0, 0, 0	2, 1, 2, 2, 2, 0, 1, 1, 1	0, 0	0, 1	1,00
0, 0, 0, 0, 0, 0, 0, 0, 0	1, 1, 2, 2, 1, 1, 1, 2, 2	0, 0	0, 1	1,00
0, 0, 0, 0, 0, 0, 0, 0, 0	1, 2, 1, 2, 2, 1, 0, 2, 0	0, 0	1, 0	1,00
0, 0, 0, 0, 0, 0, 0, 0, 0	2, 1, 0, 1, 2, 0, 2, 1, 2	0, 0	1, 0	1,00
0, 0, 0, 0, 0, 0, 0, 0, 0	1, 1, 2, 1, 1, 2, 0, 2, 2	0, 0	1, 0	1,00
0, 0, 0, 0, 0, 0, 0, 0, 0	2, 1, 2, 0, 2, 2, 1, 1, 1	0, 0	0, 1	1,00

TABELA 30
TRANSIÇÕES OBSERVADAS PELO SEGUNDO MET

Espaço de Causa (saída 1, jogar, novo jogo)		Espaço de Efeito Inicial (casa 1)		Possibilidade
Causa Inicial	Causa Final	Efeito Inicial	Efeito Final	
0, 1, 0	1, 1, 0	0	1	1,00
0, 0, 0	0, 0, 1	1	0	1,00
0, 0, 0	0, 0, 0	0	2	-
0, 0, 0	0, 0, 1	2	0	1,00

A Fig. 24 mostra uma seqüência completa de jogo e a Fig. 25 mostra diversas estratégias aprendidas pelo agente que são usadas durante o jogo. Esses resultados mostram que, para o “Jogo da Velha”, o desempenho do agente controlado por METs é praticamente o mesmo de um jogador humano experiente.

Este exemplo traz algumas considerações que são importantes para um procedimento de definição de objetivos de saída. As entradas E_{10} e E_{14} representam estados de causa não controláveis (a dimensão da saída de objetivos é menor do que a dimensão do espaço de causa). Para esses estados, o MET não tem nenhum controle senão o de esperar que os mesmos ocorram. Além disso, existem objetivos que são impossíveis de atingir como $O_1 = 1$ para $E_1 = 2$. De forma geral, podemos considerar que o segundo MET irá informar ao primeiro a possibilidade de atingir um dado objetivo, conforme mostra a Tabela 31.

TABELA 31
POSSIBILIDADE DE ATINGIR OBJETIVOS VERIFICADA PELO SEGUNDO MET

E_1	O_1		
	0	1	2
0	-	possível	sem controle
1	sem controle	-	impossível
2	sem controle	impossível	-

A definição dos objetivos negados irá basicamente evitar estados em que transições sem controle do agente levem a um estado de derrota. Para evitar um valor “2”, o MET irá “fugir” do estado “0”, definindo como objetivo o estado “1”. Se, para atingir um “objetivo negado”, for necessário que ocorram simultaneamente mais do que uma “transição sem causa observável”, esse objetivo será ignorado (não será evitado).

Neste problema, quando um objetivo de saída é definido pelo primeiro MET, em princípio todas as saídas que devem ser ativadas tendem a sê-lo. Um mecanismo externo ao agente considera apenas a primeira saída que for ativada e ignora as demais. Um novo MET poderia ser utilizado para evitar que o agente tente ativar mais de uma saída ao mesmo tempo (ele receberia um reforço negativo toda vez que o fizesse e aprenderia a ativar uma saída por vez), mas essa estruturação transcende a didática deste exemplo.

Apêndice 2

Capacidade de Armazenamento de Dados da Memória Associativa Holográfica

B.1 Introdução

Nesta seção, analisaremos a capacidade de armazenamento de informação da memória associativa holográfica. Para tal, observaremos como as variâncias dos sinais de ruído associado aos sinais recuperados se comportam com a adição de novas informações na memória.

B.2 Descrição

É conhecido que, para um vetor randômico (normalizado) de dimensão n com distribuição gaussiana de média zero, o seu desvio padrão (σ) é dado por:

$$\sigma = \frac{1}{\sqrt{n}}. \quad (\text{B2.1})$$

Consideremos então uma soma de p matrizes aleatórias, dada por:

$$\mathbf{S} = \mathbf{A}_1 \mathbf{E}(1) + \mathbf{A}_2 \mathbf{E}(2) + \cdots + \mathbf{A}_p \mathbf{E}(p). \quad (\text{B2.2})$$

Ao recuperarmos qualquer valor armazenado, teremos:

$$\mathbf{A}_i \cong \mathbf{S} \mathbf{E}^T(i) = \mathbf{A}_i + \boldsymbol{\delta}. \quad (\text{B2.3})$$

A partir das equações (B2.1) (B2.2) e (B2.3), podemos demonstrar que a matriz de

erro de saída (δ) tem média zero e desvio padrão dado por:

$$\sigma_{\delta} = \sqrt{\frac{p}{n}}. \quad (\text{B2.4})$$

Consideremos agora a probabilidade (α) do módulo de qualquer um dos elementos da matriz de erro ser maior do que um dado limite ℓ . Assim,

$$\alpha = P\{|\delta_{ij}| \geq \ell\}. \quad (\text{B2.5})$$

Portanto, considerando uma distribuição de probabilidade normal com média zero e desvio padrão σ_{δ} , tem-se:

$$\alpha = \frac{2}{\sqrt{2\pi\sigma_{\delta}^2}} \int_{\ell}^{\infty} e^{-\frac{\delta^2}{2\sigma_{\delta}^2}} d\delta. \quad (\text{B2.6})$$

Para valores de $(\sigma_{\delta}^2 / \ell^2) \ll 1$, através de (B2.6), α pode ser aproximado por:

$$\alpha \approx \frac{2\sigma_{\delta}}{\ell\sqrt{2\pi}} e^{-\frac{\ell^2}{2\sigma_{\delta}^2}}. \quad (\text{B2.7})$$

Aplicando a equação (B2.4) em (B2.7), temos:

$$\alpha \approx \frac{2\sqrt{p}}{\ell\sqrt{2\pi n}} e^{-\frac{\ell^2 n}{2p}}. \quad (\text{B2.8})$$

Considerando que, para a restauração dos dados da MAH, utiliza-se $\mu = 0$, o máximo nível de ruído admissível nesse caso é dado para $\ell \rightarrow 1$. Da expressão (B2.8), podemos então determinar, de forma aproximada, o valor de p . Assim, o número máximo de

vetores que podem ser armazenados em uma MAH de dimensão $(n \times n)$, considerando-se admissível uma probabilidade de erro de recuperação α , é dado por:

$$p \cong \frac{n}{\ln \left[\frac{2}{\alpha^2 \pi} \right] - 1}. \quad (\text{B2.9})$$

Quando a dimensão do vetor de saída (m) for menor do que a dimensão de uma coluna da matriz de memória (n), teremos a informação do vetor repetida n/m vezes nos elementos da diagonal da matriz. Assim,

$$[\mathbf{a}_{ii}] = a_q, \quad i = kq \quad \text{para:} \begin{cases} k = 1, \dots, n/m \\ q = 1, \dots, m \end{cases} \quad (\text{B2.10})$$

Portanto, para a recuperação do vetor original, n/m posições da matriz de saída deverão ser avaliadas (através da obtenção de um valor médio dessas posições) para a definição de cada elemento do vetor de saída. Dessa forma, o desvio padrão (σ_8) a ser usado em (B2.7) é agora dado por:

$$\sigma_8 = \sqrt{\frac{pm}{n^2}}. \quad (\text{B2.11})$$

Sendo, neste caso, o valor de p aproximado pela expressão (B2.12). Dessa forma, o número máximo de vetores de dimensão (m), que podem ser armazenados em uma MAH de dimensão $(n \times n)$, considerando-se admissível uma probabilidade de erro de recuperação α , é dado por:

$$p \cong \frac{n^2 / m}{\ln \left[\frac{2}{\alpha^2 \pi} \right] - 1}. \quad (\text{B2.12})$$

Esta fórmula foi validada através de vários experimentos e mostrou-se bastante conservadora, sendo a incidência de erros de recuperação observados na prática bastante inferior à probabilidade de erro de recuperação teórico (α) admissível.

Referências Bibliográficas

- [1] PLATÃO. *Platão - Diálogos. Coleção Os Pensadores*. São Paulo: Nova Cultural, 1991.
- [2] TURING, A. *Computing Machinery and Intelligence in Computers and Troughs*. New York: McGraw-Hill, 1963.
- [3] NEWELL, A.; SIMON, H. A. Computer Science as Empirical Inquiry: Symbols and Search. Reprinted. In: GARFIELD, J. L. *Foundations of Cognitive Science: The Essential Readings*. New York: Paragon House, 1990, p. 113-138.
- [4] SEARLE, J. R. Minds, Brains and Programs. *The Behavioral and Brain Sciences*, v. 3, n. 3, p. 417-458, Apr. 1980.
- [5] SEARLE, J. R. *Minds Brains and Science*. Cambridge: Harvard University Press, 1984.
- [6] SEARLE, J. R. Is the Brain's Mind a Computer Program? *Scientific American*, v. 7, n. 5, p. 26-31, Jan. 1990.
- [7] LAW D. S., Subsymbolic Functionalism and Synthetic Intelligence. *Technical Report AI94-222*, Department of Computer Sciences, University of Texas, Austin, TX, 1994.
- [8] HARNAD, S. The Symbol Grounding Problem. *Physica D*, v. 42, no. 1-3, p. 335-346, Mar. 1990.
- [9] PFEIFER, R.; VERSCHURE, P. Distributed Adaptive Control: A Paradigm for Designing Autonomous Agents. In: VARELA, F. J. and BOURGNINE, P. *Proceedings of the First European Conference on Artificial Life: Toward a Practice of Autonomous Systems*. Cambridge: MIT Press/Bradford Books, 1991, p. 21-30.

-
- [10] HAYKIN, S. *Neural Networks. A comprehensive Foundation*. IEEE PRESS, 1992.
- [11] CALÔBA, L. P. Redes Neurais, Minicurso MC2. In: CONGRESSO BRASILEIRO DE AUTOMÁTICA (9.: Set. 1992).
- [12] HECHT-NIELSEN, R. Theory of the Backpropagation Neural Network. In: IEEE INTERNATIONAL CONFERENCE ON NEURAL NETWORKS (3.: Apr. 1989: Washington), *Proceedings*. Washington, 1989. v. 1. p. 593-605.
- [13] REEKE, G. N.; SPORNS, O.; EDELMAN, G. M. Synthetic Neural Modeling: The Darwin Series of Recognition Automata. *Proceedings of the IEEE*, v. 78, n. 13, p. 1498-1530, Sep. 1990.
- [14] MIIKKULAINEN, R.; DYER, M. G. Natural Language Processing with Modular Neural Networks and Distributed Lexicon. *Cognitive Science*, v. 15, n. 17, p. 343-399, Aug. 1991.
- [15] GARCIA, O. R.; AZEVEDO, F. M.; BARRETO, J. M. Genetic Algorithms in the Optimal Choise of Neural Networks for Signal Processing. In: IEEE MIDWEST SYMPOSIUM ON CIRCUITS AND SYSTEMS (38.: Aug. 1995.: Rio de Janeiro), *Proceedings*. Rio de Janeiro, 1995. p. 1361-1364.
- [16] BELEW, R. K.; MCINERNEY, J.; SCHRAUDOLPH, N. N. Evolving Networks: Using Genetic Algorithm with Connectionist Learning. *CSE Technical Report cs90-174*, Cognitive Computer Science Research Group, Computer Science & Eng. Dept., Univ. California at San Diego, Jun. 1990.
- [17] BRADSHAW, J. *Software Agents*, MIT Press, 1997.
- [18] LEVY, S. *Artificial Life*, New York: Vintage Books, 1993.
- [19] BROOKS, R.; FLYN, A. M. Fast, Cheap and aut of Control, *Journal of the British Interplanetary Society*, v. 42, n. 12, p. 478-485, Nov. 1989.
- [20] BROOKS, R. A. Intelligence without Representation, *Artificial Intelligence Journal*, v. 47, n. 17, p. 139-160, Sep. 1991.
- [21] ASIMOV, I. *Eu Robô*, São Paulo: Círculo do Livro, 1998.

-
- [22] KAKU, M. *Visões de Futuro: Como a Ciência Revolucionará o Século XXI*. Rio de Janeiro: Rocco, 2001.
- [23] PIERCE, C. S. *Semiótica e Filosofia. Textos Escolhidos de Charles S. Pierce*. São Paulo: Cultrix/USP, 1975.
- [24] PIAGET, J. A. *Equilibração das Estruturas Cognitivas - Problema Central do Desenvolvimento*. Rio de Janeiro: Zahar, 1976.
- [25] MORRIS, C. W. *Foundation for a Theory of Signs. Writings on the General Theory of Signs*. Mouton: The Hague, 1971.
- [26] PIAGET, J.; INHELDER, B. *Gênese das Estruturas Lógicas Elementares*. Rio de Janeiro: Zahar, 1976.
- [27] GUDWIN, R. R. *Contribuições ao Estudo Matemático de Sistemas Inteligentes*. Campinas, 1996. Tese (Doutorado em Matemática Estatística e Computação Científica), UNICAMP.
- [28] PIAGET, J.; INHELDER, B. *Memory and Intelligence*. London: Routledge & Kegan Paul, 1973.
- [29] BREARLEY, M.; HITCHFIELD, E. *Guia Prático Para Entender Piaget*. São Paulo: Ibrasa, 1976.
- [30] DAVIDSSON, P. On the Concept of Concept in the Context of Autonomous Agents. In: WORLD CONFERENCE ON THE FUNDAMENTALS OF ARTIFICIAL INTELLIGENCE (2.: Jul. 1995, Paris, France), *Proceedings*, Paris, France, 1995. p. 85-96.
- [31] FIALHO, F. A. P. *Modelagem Computacional das Estruturas Cognitivas Conforme Proposto por Jean Piaget*. Florianópolis, 1993. Tese (Doutorado em Engenharia de Produção) - Centro Tecnológico, Universidade Federal de Santa Catarina.
- [32] ULIANA, P. B.; SEARA, R.; BERMUDEZ, J. C. M. Escalas de Medição dos Graus de Adaptabilidade e Inteligência de Agentes Cognitivos. In: CONGRESSO BRASILEIRO DE REDES NEURAIIS (4.: Jul. 1999: São José dos Campos, SP). *Anais*. São José dos Campos, 1999. p. 367-372.

-
- [33] WATKINS, C. J. C. H.; DAYAN P. Q-learning. *Machine Learning*, v. 8, n. 3, p. 279-292, Mar. 1992.
- [34] BARTO, A. G.; SUTTON, R. S.; ANDERSON, C. W. Neuronlike Adaptive Elements that Can Solve Difficult Learning Control Problems. *IEEE Trans. on Systems, Man*, v. 25, n. 21, p. 234-240, 1983.
- [35] DAYAN, P.; SEJNOWSKI, T. J. TD(λ) Converges with Probability 1. *Machine Learning*, v. 5, n. 7, p. 114-117, Aug. 1994.
- [36] SUTTON, R. S. *Temporal Credit Assignment in Reinforcement Learning*. Amherst, 1984. Ph.D. Thesis, Department of Computer Science, University of Massachusetts.
- [37] ACKLEY, D. H.; LITTMAN, M. L. Generalization and Scaling in Reinforcement Learning. In TOURETZKY, D. S. *Advances in Neural Information Processing Systems 2*, San Mateo: Morgan Kaufmann, 1990, p. 550-557.
- [38] KAEHLING, L. P.; LITTMAN, M. L.; MOORE, A. W. Reinforcement Learning: A Survey, *Journal of Artificial Intelligence Research*, v. 93, n. 4, p. 87-112, Apr. 1996.
- [39] HAUSKRECHT, M.; MEULEAU, N., BOUTILIER, C. *et al.* Hierarchical Solution of Markov Decision Processes Using Macro-Actions. In: INTERNATIONAL CONFERENCE ON UNCERTAINTY IN ARTIFICIAL INTELLIGENCE (14.: Aug. 1998: San Francisco, USA). *Proceedings*, San Francisco, USA, 1998. p. 315-321.
- [40] BALKENIUS, C. *Neural Mechanisms for Self- organization of Emergent Schemata, Dynamic Schema Processing, and Semantic Constraint Satisfaction*. Lund University Cognitive Studies, 14, 1992
- [41] SUN, R.. On Schemas, Logics, and Neural Assemblies. *Applied Intelligence*, v. 5, n. 2, p. 83-102, Mar. 1995G.
- [42] DRESCHER, G. L. *Made-up Minds - A Construtivist Approach to Artificial Intelligence*. Massachusetts: The MIT Press, 1991.

- [43] WAZLAWICK, R. S. *Um Modelo Operatório para Construção de Conhecimento*. Florianópolis, 1993. Tese (Doutorado em Engenharia de Produção) – Centro Tecnológico, Universidade Federal de Santa Catarina.
- [44] ULIANA, P. B.; SEARA, R.; BERMUDEZ, J. C. M. Treinamento Condicionado Através de Esquemas de Transição. In: CONGRESSO BRASILEIRO DE REDES NEURAIS (4.: Jul. 1999: São José dos Campos, SP). *Anais*. São José dos Campos, 1999. p. 361-366.
- [45] PLATE, T. Holographic Reduced Representations. *IEEE Trans. on Neural Networks*, v. 6, n. 4, p. 623-641, Apr. 1995.
- [46] POLLACK, J. B. Recursive Distributed Representations. *Artificial Intelligence*, v. 6, n. 46, p. 77-105, Jan. 1990.
- [47] KOSKO, B. Bidirecional Associative Memories. *IEEE Trans. on Systems, Man, and Cybernetics*, v. 18, n. 1, p. 49-60, Jan./Feb., 1988.
- [48] HAYKIN, S. *Neural Networks. A Comprehensive Foundation*. Macmillan College Publishing Company, 1994.
- [49] HUNT, B.; NADAR, M. S.; GOYAL, A. Synthesis of a Nonrecurrent Associative Memory Model Based on a Nonlinear Transformation in the Spectral Domain, *IEEE Trans. on Neural Networks*, v. 4, n. 5, p. 873-878, Sep. 1993.
- [50] ADAMSON, M. J.; DAMPER, R. I. A Recurrent Network which Learns to Pronounce English Text. In: INTERNATIONAL CONFERENCE ON SPOKEN LANGUAGE PROCESSING (2.: Mar. 1996: Philadelphia). *Proceedings*. Philadelphia, 1996. v. 3, p. 1704-1707.
- [51] KOHONEN, C. S. T. Correlation Matrix Memories, *IEEE. Trans. on Computer*, v. C-21, n. 32, p. 353, Oct. 1972.
- [52] POGGIO, T. On Optimal Nonlinear Associative Recall, *Biol. Cyb.*, v. 19, n. 15, p. 201-209, May 1975.
- [53] POLLACK, J. B. Recursive Auto-Associative Memory: Devising Compositional Distributed Representations, In: ANNUAL CONF. OF THE COGNITIVE SCIENCE SOCIETY (10.: Mar. 1988: Montreal). *Proceedings*. Montreal, 1988. p. 33-39.

-
- [54] PLATE, T. *Distributed Representations and Nested Compositional Structure*. Toronto, 1994. Ph.D. Thesis, Department of Computer Science, University of Toronto.
- [55] ULIANA, P. B.; SEARA, R.; BERMUDEZ, J. C. M. Memória Associativa Holográfica. In: CONGRESSO BRASILEIRO DE REDES NEURAIAS (5.: Abr. 2001: Rio de Janeiro). *Anais*. Rio de Janeiro, 2001. p. 349-354.
- [56] MINSKY, M. A. Framework for Representing Knowledge. In: COLLINS, A.; SMITH, E. E. *Readings in Cognitive Sciences, A Perspective from Psychology and Artificial Intelligence*, Morgan Kaufmann Publishers, 1988.
- [57] BEAUCHAMP, K. G., *Application of Walsh and Related Function*, Academic Press, 1984.
- [58] ZADEH, L. A. Fuzzy Sets as a Basis for a Theory of Possibility, *Fuzzy Sets and Systems*, v. 1, 1978, p. 3-28.