3
Reinforcement Learning Neuro-Fuzzy Hierárquico Politree (RL-NFHP)

3.1
Introdução

Este capítulo apresenta formalmente o modelo Reinforcement Learning Neuro-Fuzzy Hierárquico Politree (RL-NFHP) proposto em Figueiredo (2003). O particionamento Politree é uma generalização sobre a forma de particionamento Quadtree, que por sua vez pode ser visto como uma extensão do particionamento binário do espaço (BSP).

3.2
Particionamento Quadtree/Politree

No particionamento Quadtree o espaço é sucessivamente dividido em quadrantes, que, por sua vez podem ser novamente subdivididos em 4 regiões (quadrantes) em uma operação recursiva.

A figura 7a ilustra este tipo de particionamento para o caso de duas dimensões, e a figura 7b mostra a representação da árvore Quadtree.

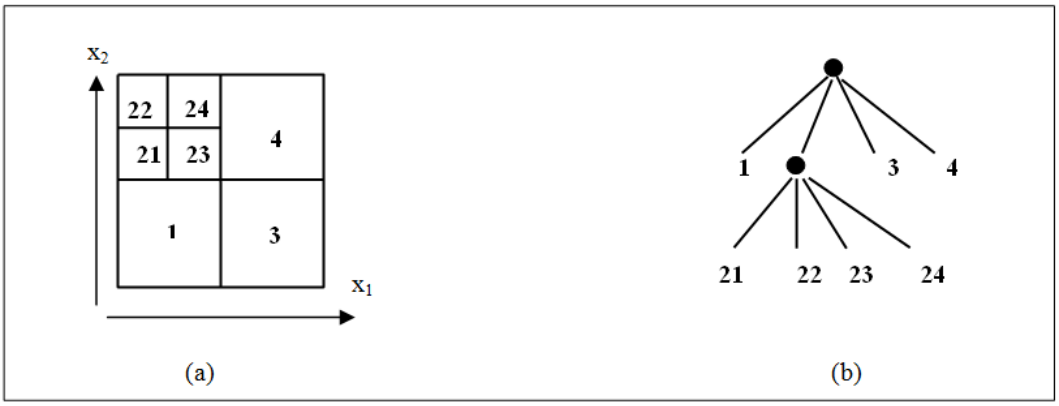

Figura 7: (a) exemplo de particionamento Quadtree; (b) árvore representativa do particionamento Quadtree referente a Figura 7ª.

O particionamento Quadtree pode ser fixo, como mostrado na Figura 7a, ou adaptativo. Neste último caso, as regiões geradas em cada subdivisão são retangulares e não mais quadradas, como ocorre no caso do particionamento fixo. A limitação do particionamento Quadtree (fixo ou adaptativo) está no fato de este trabalhar apenas em espaços bidimensionais. Isto pode ser contornado pela extensão para casos n-dimensionais. Por exemplo, no caso de dimensão n=3, temos o particionamento "Oct-tree" (Tamminen, 1984), (Arvo, 1988) que divide o espaço em 8 (2^3) subespaços. Para n = 4 o espaço (hiperespaço) seria subdividido em 16 (2^4) subespaços e assim por diante. O particionamento utilizado no segundo modelo proposto nesta tese denominado *Reinforcement Learning* – Neuro-Fuzzy Hierárquico Politree (RL-NFHP) é uma generalização desta idéia. Nele, a subdivisão do espaço dimensional é realizada em n subespaços.

O modelo RL-NFHP é composto de uma ou várias células padrão chamadas *células RL-neuro-fuzzy politree* (RL-NFP). Estas células são dispostas numa estrutura hierárquica em forma de árvore. A célula de maior hierarquia gera a saída. As de menor hierarquia trabalham como conseqüentes das células de maior hierarquia. Estas células são descritas em detalhes na seção 3.3, a seguir.

3.3
Célula Básica RL-Neuro-Fuzzy Politree

Uma célula RL-NFP é um mini-sistema neuro-fuzzy que realiza um particionamento politree em um determinado espaço, utilizando, para cada variável de entrada, as funções de pertinência descritas na seção 3.3. A célula RL-NFP gera uma saída precisa (*crisp*) após um processo de defuzzificação, conforme será mostrado posteriormente.

Apenas para efeito de ilustração, na representação da célula serão apresentadas duas entradas (Quadtree) tornando o desenho mais simples do que a forma n-dimensional proposta para o Politree.

As figuras 8 e 9, a seguir, foram criadas para facilitar a compreensão do processo de defuzzificação da célula e o encadeamento dos conseqüentes. As entradas x_1 e x_2 geram os antecedentes das quatro regras fuzzy após serem computados os graus de pertinência $\rho_1(x_1)$, $\mu_1(x_1)$, $\rho_2(x_2)$ e $\mu_2(x_2)$, onde: ρ_1 é o

conjunto nebuloso *baixo* e μ_1 é o conjunto nebuloso *alto* relativos à entrada x_1; e ρ_2 é o conjunto nebuloso *baixo* e μ_2 é o conjunto nebuloso *alto* relativos à entrada x_2. Os valores definidos como conseqüentes são conjuntos de ações (a_1, a_2, ... a_t), onde cada ação está associada a uma função de valor-Q. Através do método de aprendizado baseado em RL, uma ação de cada polipartição (a_i, a_j, a_p e a_q) será definida como aquela que representa o comportamento desejado do sistema quando o mesmo se encontra em um determinado estado. A figura 8 ilustra a representação desta célula de forma simplificada e a figura 9 apresenta-a sob o formato de rede neuro-fuzzy.

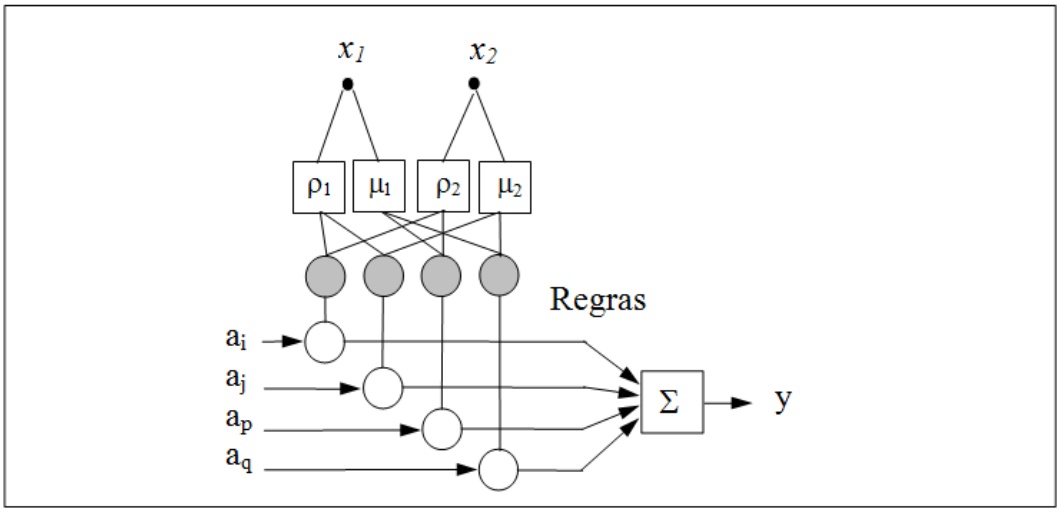

Figura 8: Célula Reinforcement Learning Neuro-Fuzzy Quadtree (Politree com n=2).

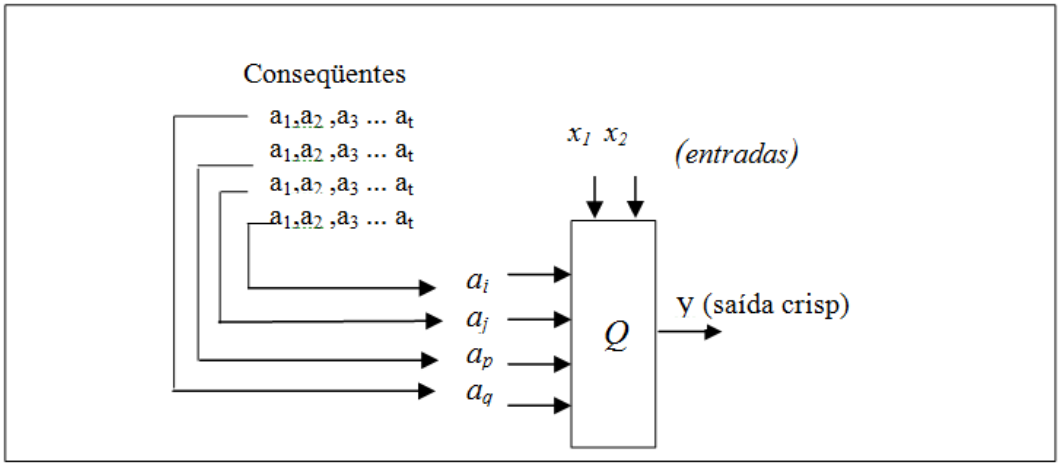

Figura 9: Diagrama simplificado da célula Reinforcement Learning Neuro-Fuzzy Quadtree relativo a figura 8.

Na figura 10, estão mostradas as camadas de fuzzificação, de regras e de defuzzificação. As expressões analíticas das funções de pertinência *alto* e *baixo* são dadas por sigmóides e seus complementos de 1. Estas funções de pertinência (FPs) possuem 2 parâmetros, 'a' e 'b', que definem os perfis das funções alto (μ) e baixo (ρ) de cada variável de entrada. Os α_i (na figura 10) simbolizam os graus de ativação das regras. Estes graus de ativação são calculados usando-se uma operação AND (T-norma) sobre os graus de pertinência de ρ_1, μ_1, ρ_2 e μ_2, conforme descrito a seguir:

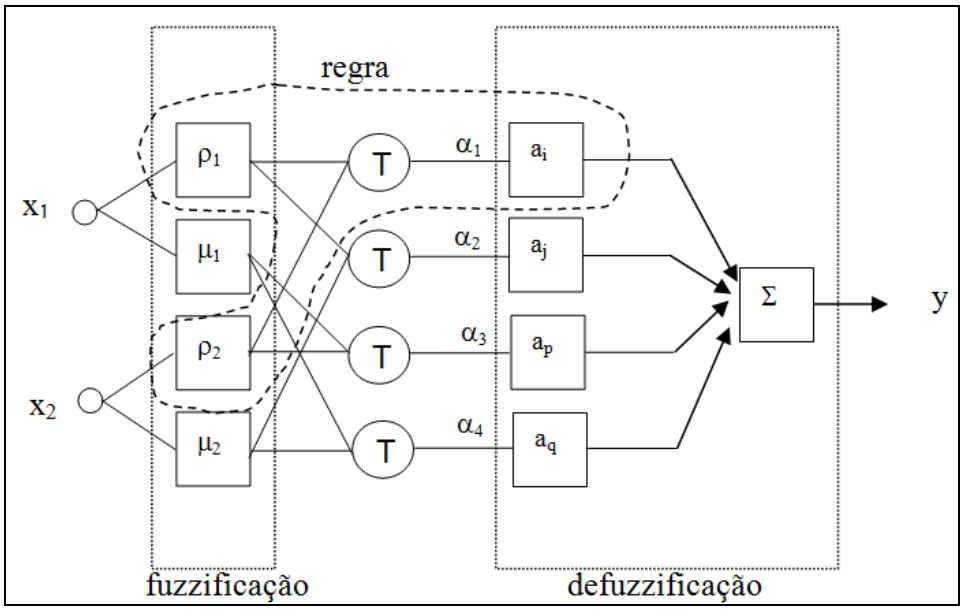

Figura 10: Célula RL-NFP representada sob o formato de rede neuro-fuzzy.

$$\begin{aligned} \alpha_1 &= \rho_1(x_1) * \rho_2(x_2); \\ \alpha_2 &= \rho_1(x_1) * \mu_2(x_2); \\ \alpha_3 &= \mu_1(x_1) * \rho_2(x_2); \\ \alpha_4 &= \mu_1(x_1) * \mu_2(x_2). \end{aligned} \tag{15}$$

O símbolo '*' representa a operação AND, que pode ser realizada, por exemplo, pela multiplicação ou pela operação de mínimo entre os dois valores. A interpretação lingüística do mapeamento implementado pela célula RL-NFP da figura 10 é dada pelo seguinte conjunto de regras:

regra$_1$: Se $x_1 \in \rho_1$ e $x_2 \in \rho_2$ então $y = a_i$

regra$_2$: Se $x_1 \in \rho_1$ e $x_2 \in \mu_2$ então $y = a_j$

regra$_3$: Se $x_1 \in \mu_1$ e $x_2 \in \rho_2$ então $y = a_p$

regra$_4$: Se $x_1 \in \mu_1$ e $x_2 \in \mu_2$ então $y = a_q$

Cada regra corresponde a um quadrante da figura 11. Quando os valores das entradas incidem sobre o quadrante 1, é a regra 1 que tem maior grau de ativação. Quando a incidência é sobre o quadrante 2, é a regra 2 que tem maior grau de ativação. No caso das entradas caírem no quadrante 3, é a regra 3 que tem o maior grau de ativação e, finalmente, quando a incidência é sobre o quadrante 4, é a regra 4 que tem maior grau de ativação. Cada quadrante por sua vez pode ser subdividido em quatro partes, através de uma outra célula RL-NFP. É muito importante lembrar que os conseqüentes a_i não são valores predeterminados, eles fazem parte de um conjunto de ações que deve ser explorado para que se possa determinar, através de aprendizado por reforço, a ação mais adequada para cada regra.

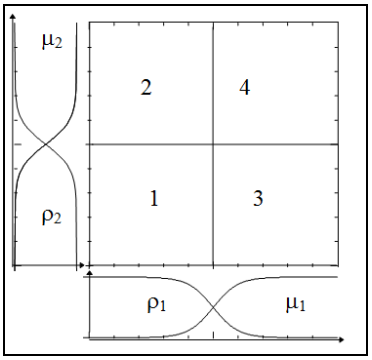

Figura 11: Divisão em quadrantes realizada pelas FPs alto e baixo.

A célula *RL-neuro-fuzzy politree* pode ser representada por uma estrutura em árvore como mostra a figura 12a. Ela corresponde ao particionamento da figura 11 com duas entradas. Este formato proporciona a representação da célula RL-NFP genérica (com n entradas) que é exibida na figura 12b.

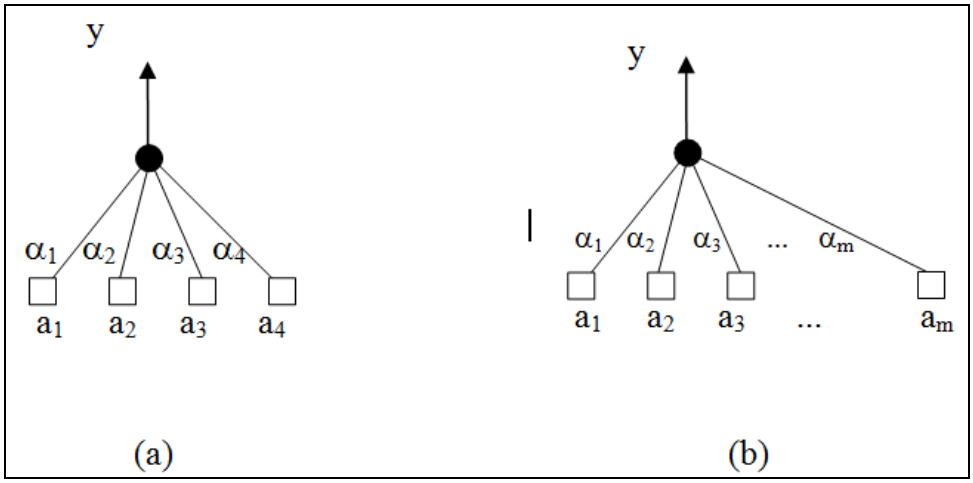

Figura 12: (a) representação em árvore da célula RL-NFP com duas entradas; (b) representação genérica em árvore da célula RL-NFP com n entradas, onde m = 2^n.

As saídas 'y' das células RL-NFP das figuras 12a e 12b são dadas pelas médias ponderadas mostradas nas equações 16 e 17, respectivamente:

$$y = \left(\frac{\sum_{i=1}^{4} \alpha_i \times a_i}{\sum_{i=1}^{4} \alpha_i} \right) \tag{16}$$

$$y = \left(\frac{\sum_{i=1}^{2^n} \alpha_i \times a_i}{\sum_{i=1}^{2^n} \alpha_i} \right) \tag{17}$$

onde n é o número de entradas e a_i corresponde a um dos dois conseqüentes possíveis abaixo:

- *um singleton* (conseqüente *fuzzy singleton,* ou Sugeno de ordem zero), caso em que a_i = constante;
- *saída de um estágio de nível anterio*r, caso em que $a_i = y_j$, onde y_j representa a saída de uma célula genérica 'j', cujo valor é calculado, também, pela eq. 17.

Apesar do conseqüente *singleton* ser simples, este não é conhecido previamente. É através do algoritmo RL que será possível determinar o melhor valor *singleton* (ação) para esta regra. Ou seja, os conseqüentes de cada regra são representados por um conjunto de ações relacionadas àquele estado. O estado atual do agente é definido pelos valores das variáveis de entrada, que tornam ativas as células cujos domínios das funções de pertinência delimitam um estado.

Como mencionado, na célula RL-NFP básica as funções de pertinência são implementadas por *sigmóides* (ρ e μ) e por seu complemento de um [1- $\mu(x)$]. A utilização dos complementos a um leva a uma simplificação no procedimento de defuzzificação realizado pelo processo de média ponderada, pois o somatório dado pela equação abaixo, é igual a 1 para quaisquer valores de entrada x_i.

$$\sum_{i=1}^{2^n} \alpha_i = 1 \tag{18}$$

Desta forma, a saída da célula básica (eq. 17) fica simplificada, como mostra a eq. 19, a seguir.

$$y = \sum_{i=1}^{2^n} \alpha_i . a_i \tag{19}$$

As células RL-NFP formam uma estrutura hierárquica que resulta nas regras que compõem o raciocínio do agente (seção 2.2.2). Para este modelo, os antecedentes das regras são definidos pelas variáveis de entrada, as quais estão associados dois conjuntos fuzzy. No caso deste modelo, todas as variáveis de entrada do sistema compõem os antecedentes das células. Os valores das variáveis de entrada são lidos pelos ***sensores*** do agente e são os respectivos graus de pertinência são determinados pelos antecedentes. Se o grau de ativação não é nulo (resultado da aplicação do T-norma é diferente de zero), a regra é disparada. Os conseqüentes são as ações que o agente deve aprender ao longo do processo e são realizadas pelo seu ***atuador***. Sendo assim, o modelo RL-NFHP também cria e determina sua estrutura mapeando estados em ações.

3.4
Arquitetura RL-NFHP

A arquitetura do modelo RL-NFHP é composta pela interligação entre as células básicas descritas acima. Isto é exemplificado na figura 13, abaixo. A árvore *Politree* referente ao particionamento da figura 13 é mostrada na figura 14. Cada partição não subdividida é chamada de *polipartiçã*o.

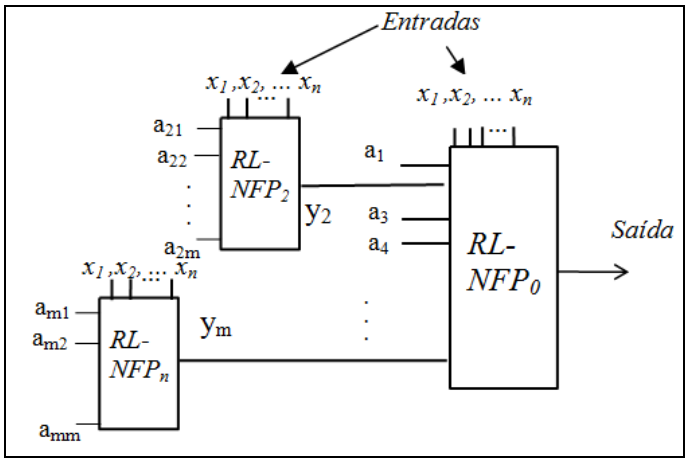

Figura 13: Exemplo de arquitetura RL-NFHP.

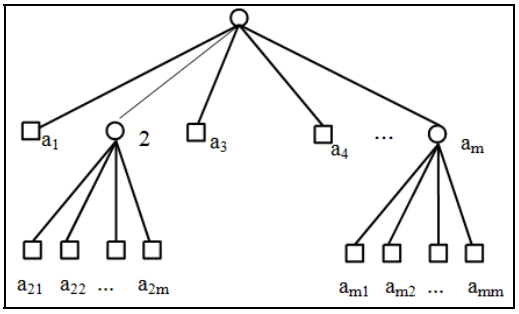

Figura 14: Árvore *Politree* referente ao particionamento da figura 13.

Na árvore da figura 14, os nós simbolizados com pequenos círculos são nós interiores e representam regiões que foram subdivididas. Os nós simbolizados por pequenos quadrados são nós terminais e representam as polipartições, isto é, as regiões que não sofreram subdivisões. A raiz da árvore simboliza todo o espaço a ser particionado.

No exemplo das figuras 13 e 14 as polipartições 1, 3, 4 não foram subdivididas, portanto os conseqüentes de suas respectivas regras são os valores a_1, a_3 e a_4. As partições 2 e m foram subdivididas e os conseqüentes de suas regras são as saídas (y_2 e y_m) dos subsistemas 2 e n. Estes por sua vez têm, como conseqüentes, os valores a_{21}, a_{22}, ..., a_{2m}, e a_{m1}, a_{m2}, ..., a_{mm}, respectivamente. Cada *'a_i'* corresponde a um conseqüente de *Sugeno* de ordem 0 (*singleton*), representando a ação que será identificada (dentre as ações possíveis), através de aprendizado por reforço, como sendo a mais favorável para um determinado estado do ambiente. A saída do sistema da figura 13 é dada pela eq. 20, a seguir.

$$y = \alpha_1.a_1 + \alpha_2 \sum_{i=1}^{2^n} \alpha_{2i}.a_{2i} + \alpha_3.a_3 + \alpha_4.a_4 + \ldots + \alpha_m \sum_{i=1}^{2^n} \alpha_{mi}.a_{mi} \qquad (20)$$

De uma forma genérica, a equação de saída de um sistema RL-NFHP de dois níveis completos é dada pela eq. 21 abaixo. Neste caso houve necessidade de se incluir as variáveis k_i e k_{ij} . Essas variáveis assumem apenas valores iguais a '0' ou '1', indicando a existência ou não das polipartições de ordem *'i'* e *'ij*', respectivamente.

$$y = \sum_{i=1}^{2^n} \alpha_i k_i a_i + \sum_{i=1}^{2^n} \sum_{j=1}^{2^n} \alpha_i \alpha_{ij} k_{ij} a_{ij} \qquad (21)$$

Expandindo a equação 21 para um sistema RL-NFHP de quatro níveis de hierarquia tem-se a seguinte fórmula:

$$\begin{aligned} y = & \sum_{i=1}^{2^n} \alpha_i k_i a_i + \\ & \sum_{i=1}^{2^n} \sum_{j=1}^{2^n} \alpha_i \alpha_{ij} k_{ij} a_{ij} + \\ & \sum_{i=1}^{2^n} \sum_{j=1}^{2^n} \sum_{p=1}^{2^n} \alpha_i \alpha_{ij} \alpha_{ijp} k_{ijp} a_{ijp} + \\ & \sum_{i=1}^{2^n} \sum_{j=1}^{2^n} \sum_{p=1}^{2^n} \sum_{q=1}^{2^n} \alpha_i \alpha_{ij} \alpha_{ijp} \alpha_{ijpq} k_{ijpq} a_{ijpq} \end{aligned} \qquad (22)$$

Na equação 22:

- α_i , α_{ij} , α_{ijp} , α_{ijpq} , são os níveis de disparo das regras de cada polipartição i, ij, ijk, ou ijkl, respectivamente;
- k_i (k_{ij} , k_{ijp} , k_{ijpq}), é igual a "1" se a partição i (ou ij, ou ijp ou ijpq) existe e "0" caso contrário;
- a_i , a_{ij} , a_{ijp} , a_{ijpq} , são os conseqüentes (*singletons*) das regras existentes.

Na equação da expressão geral de saída do modelo RL-NFHP, descrita acima, já se levou em consideração a simplificação causada pelo uso das funções de pertinência complementares ($\rho + \mu = 1$) no método de defuzzificação das saídas de cada subsistema neuro-fuzzy.

O conjunto de regras que traduz o conhecimento lingüístico do exemplo da figura 13 é:

Se $x_1 \in \rho_1$ e $x_2 \in \rho_2 \ldots x_n \in \rho_n$ então $y = a_1$
Se $x_1 \in \mu_1$ e $x_2 \in \rho_2 \ldots x_n \in \rho_n$ então
{
Se $x_1 \in \rho_{21}$ e $x_2 \in \rho_{22} \ldots x_n \in \rho_{2n}$ então $y = a_{21}$
Se $x_1 \in \mu_{21}$ e $x_2 \in \rho_{22} \ldots x_n \in \rho_{2n}$ então $y = a_{22}$
Se $x_1 \in \mu_{21}$ e $x_2 \in \mu_{22} \ldots x_n \in \rho_{2n}$ então $y = a_{23}$
.
.
Se $x_1 \in \mu_{21}$ e $x_2 \in \mu_{22} \ldots x_n \in \mu_{2n}$ então $y = a_{2m}$
}
Se $x_1 \in \mu_1$ e $x_2 \in \mu_2 \ldots x_n \in \rho_n$ então $y = a_3$
Se $x_1 \in \mu_1$ e $x_2 \in \mu_2 \ldots x_n \in \rho_n$ então $y = a_4$
.
.
Se $x_1 \in \mu_1$ e $x_2 \in \mu_2 \ldots x_n \in \mu_n$ então
{
Se $x_1 \in \rho_{m1}$ e $x_2 \in \rho_{m2} \ldots x_n \in \rho_{mn}$ então $y = a_{m1}$
Se $x_1 \in \mu_{m1}$ e $x_2 \in \rho_{m2} \ldots x_n \in \rho_{mn}$ então $y = a_{m2}$
Se $x_1 \in \mu_{m1}$ e $x_2 \in \mu_{m2} \ldots x_n \in \rho_{mn}$ então $y = a_{m3}$
.
.
Se $x_1 \in \mu_{m1}$ e $x_2 \in \mu_{m2} \ldots x_n \in \mu_{mm}$ então $y = a_{mm}$
}

Onde:

- $\rho_1, \rho_2, \ldots, \rho_n$ e $\mu_1, \mu_2, \ldots, \mu_n$, são as funções de pertinência que definem a partição de nível 1.

- $\rho_{21}, \rho_{22},\ldots, \rho_{2n}$ e $\mu_{21}, \mu_{22}, \ldots, \mu_{2n}$, são as funções de pertinência que definem as subdivisões da partição 2.

- $\rho_{m1}, \rho_{m2},\ldots, \rho_{mn}, \mu_{m1}, \mu_{m2}, \ldots, \mu_{mn}$, são as funções de pertinência que definem as subdivisões da partição m.

3.4.1
Antecedentes das Regras do Modelo RL-NFHP

A figura 15 mostra a estrutura de aprendizado do agente. A leitura do ambiente ($s_1, s_2, ..., s_n$) é feita pelo agente através de seus sensores e estas leituras podem ser traduzidas em um ou mais valores de entrada ($x_1, x_{2, \ldots} x_n$) das células; no entanto, cada célula tem associados a ela **todos** as entradas que serão considerados no sistema no momento de sua criação. Os valores x_i são avaliados nas células, podendo disparar regras Sendo assim, toda vez que uma regra é disparada, ou, dito de outra forma, uma **célula** se torna **ativa**, o processo de aprendizado identifica que o agente está em um **estado definido** pelo domínio dos conjuntos fuzzy do antecedente da regra (domínio da entrada da célula).

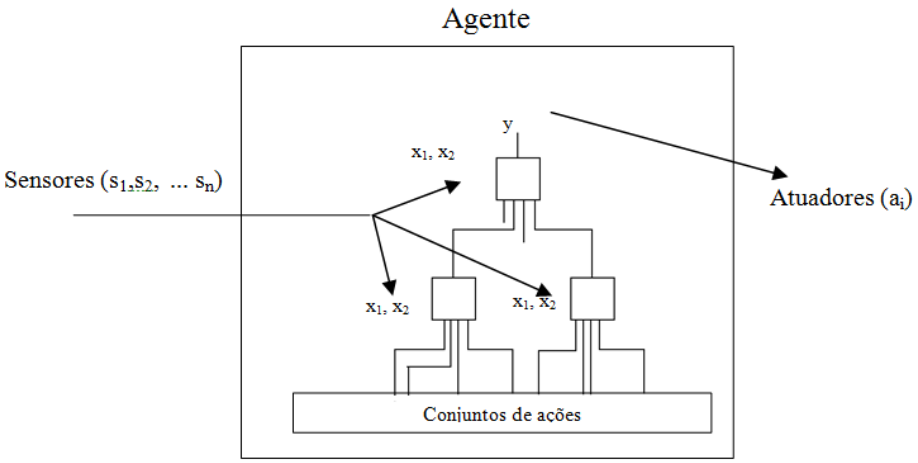

Figura 15: Esquema do processo de aprendizado do agente.

3.4.2
Conseqüentes das Regras do Modelo RL-NFHP

Os conseqüentes das regras fuzzy nas células RL-NFHP são as ações que devem ser identificadas através de aprendizado por reforço. Quando as células se tornam ativas, as ações são selecionadas, cada uma relativa à combinação *baixo* e

alto de cada uma das entradas da célula. A seleção ocorre em função de valores atribuídos a cada uma das ações que pertencem ao conjunto de ações disponíveis para cada polipartição *baixa* e *alta*.

Novamente, para efeito de ilustração (figura 16), serão apresentadas duas entradas (Quadtree), tornando o desenho mais simples do que a forma n-dimensional proposta para o Politree. A figura 16 mostra a célula RL-NFHP com 4 conjuntos de ações (associados aos seus respectivos Q-valores), onde cada conjunto está relacionado às polipartições de cada célula. Cada conjunto pode possuir um número de ações t, independentemente do número de entradas.

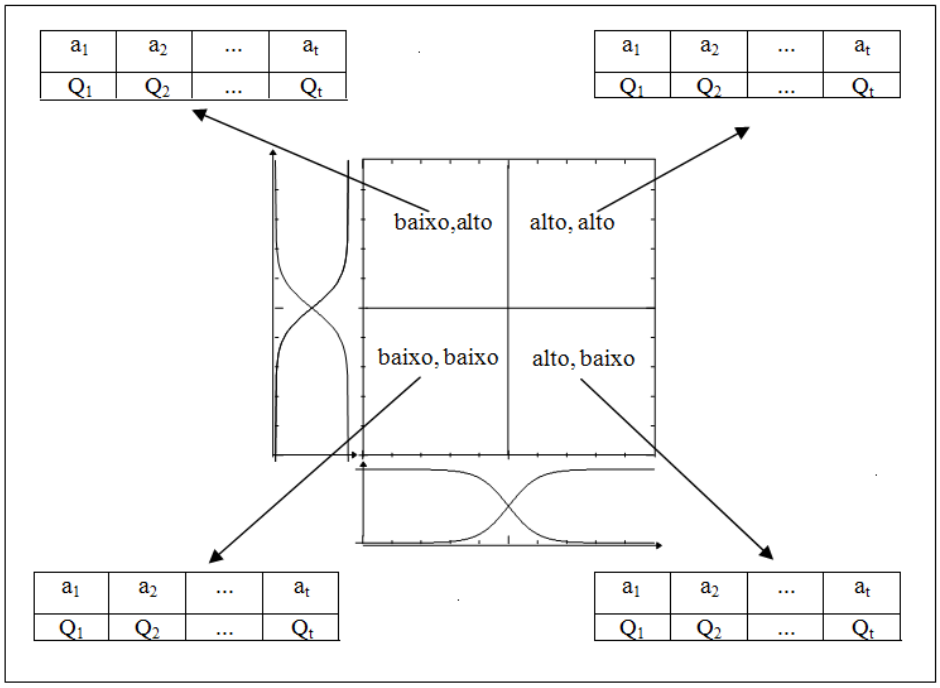

Figura 16: Interior da célula RL-NFHP com duas entradas (Quadtree).

Os conseqüentes serão do tipo 1 se a célula for uma folha da estrutura e serão do tipo 2 se forem células intermediárias.

Conseqüente Tipo 1 – Singleton

É o tipo mais simples de conseqüente de uma regra fuzzy. As regras fuzzy com este conseqüente são mostradas abaixo.

Se $x_1 \in \rho_1$ e $x_2 \in \rho_2$ então $y = a_i$

Se $x_1 \in \rho_1$ e $x_2 \in \mu_2$ então $y = a_j$

Se $x_1 \in \mu_1$ e $x_2 \in \rho_2$ então $y = a_p$

Se $x_1 \in \mu_1$ e $x_2 \in \mu_2$ então $y = a_q$

A vantagem do uso deste tipo de conseqüente está na facilidade do cálculo da saída que, neste caso, é geralmente efetuado através da média ponderada. Este tipo de conseqüente também é conhecido como conseqüente de *Sugeno* de ordem '0'.

Apesar de o método de cálculo para este tipo de conseqüente ser simples, o valor do conseqüente **não é conhecido a priori**. Um dos objetivos do modelo é, portanto, aprender, através do algoritmo SARSA (Sutton e Barto, 1998), a identificar as ações associadas aos conjuntos fuzzy baixo e alto da célula RL-NFP que melhor respondam ao estado atual do agente.

Conseqüente Tipo 2 – Célula RL- NFP

Este tipo de conseqüente é, na verdade, a saída de um outro mini-sistema RL- Neuro-Fuzzy Hierárquico Politree implementado por uma célula RL-NFP. Isto gera a hierarquia inerente aos sistemas neuro-fuzzy hierárquicos. As regras fuzzy com este conseqüente são como as que foram mostradas na seção 3.4- Arquitetura RL-NFHP.

3.5
Algoritmo de Aprendizado

O processo de aprendizado começa com a definição das entradas relevantes, para o sistema/ambiente no qual o agente está inserido, e dos conjuntos de ações que ele pode dispor para atingir seus objetivos. As funções de valores-Q iniciais associadas às ações também devem ser definidas. Normalmente, as aplicações que utilizam SARSA ou *Q-Learning* iniciam suas funções de valores-Q com zero.

O algoritmo de aprendizado ocorre em seis fases, que são descritas a seguir.

Passos do Aprendizado:

1 - Inicialização

Uma célula raiz é criada, tendo como domínios dos seus conjuntos fuzzy (como definidos na seção 3.3), relativos a cada uma das entradas, os valores mínimo e máximo destas entradas (Limite Inferior – LI e Limite Superior – LS). Com o objetivo de generalidade, os valores das variáveis de entrada são normalizados. Os valores correspondentes às variáveis de entrada da célula são lidos do ambiente, normalizados e podem ser aplicados diretamente às entradas da célula ou ser modificados segundo uma função que os torne adequados às variáveis das entradas da célula. Estes valores são avaliados nos conjuntos fuzzy baixo e alto, resultando nos graus de pertinência $\rho(x_1)\mu(x_1)$, ... $\rho(x_n)\mu(x_n)$, respectivamente, para cada variável de entrada. Cada uma das polipartições escolhe uma das ações de seu conjunto de ações baseando-se nos métodos descritos na seção 3.5 (passo 3 do algoritmo – Seleção das ações). A saída da célula é calculada pelo processo de defuzzificação e representa a ação que será executada pelos *atuadores* do agente.

Com a utilização de mais entradas, pode ocorrer que os valores α_i (resultado da aplicação da T-norma sobre os graus de pertinência relativos às entradas da célula) se tornem muito pequenos, o que faria o algoritmo ter um gasto maior de tempo computacional executando cálculos para a saída e atualizações das funções de valor que não teriam um peso significante para o algoritmo. Sendo assim, foi acrescentado o conceito de *alfa-cut* (limiar de corte) com o objetivo de inibir na saída ações relacionadas a polipartições cujos α_i sejam muito pequenos. Neste caso, o valor α_i desta polipartição torna-se igual a zero. Isso significa que, nesta iteração, esta polipartição não contribuirá na saída com sua ação, nem terá a sua função de valor-Q (associada à ação selecionada) atualizada.

2 - Função de Avaliação/Retorno

Após a execução da ação, uma nova leitura do ambiente é realizada. Esta leitura permite que seja calculado o valor de reforço do ambiente e se avalie a ação tomada pelo agente. Este valor deve ser calculado através de uma função de avaliação definida segundo os objetivos do agente, sendo fundamental para a orientação do agente ao longo do processo de aprendizado.

3 - Retropropagar o retorno

A cada passo, no processo de aprendizado, o retorno é calculado para cada partição de todas as células ativas, mediante a sua participação na ação resultante. Dessa forma, o retorno do ambiente é retropropagado a partir da célula raiz até as células folhas, conforme o exemplo do sistema da figura 17. A figura 17a mostra duas células com duas entradas. Sendo assim, cada célula possui 4 polipartições, cada uma relativa à combinação baixo/alto dos graus de pertinência avaliados pelas entradas x_1 e x_2. A figura 17b mostra o retorno global do sistema representado pela letra R no topo da árvore, que será definido conforme a aplicação ou estudo de caso em questão. Os valores correspondentes aos retornos de cada partição da célula RL-NFP_0 são $R_{0bb,}$ R_{0ba} , R_{0ab} , R_{0aa}, e são calculados mediante os graus de pertinências correspondentes aos α_i relativos à célula RL-NFP_0.

$R_{0bb,}$ R_{0ba} , R_{0ab} , R_{0aa} são os valores dos retornos locais da célula ‘0’ calculados para as polipartições bb (baixo/baixo), ba (baixo/alto), ab (alto/baixo) e aa (alto/alto) desta célula.

$R_{1bb,}$ R_{1ba} , R_{1ab} , R_{1aa} são os valores dos retornos locais da célula ‘1’calculados para as polipartições bb (baixo/baixo), ba (baixo/alto), ab (alto/baixo) e aa (alto/alto) desta célula.

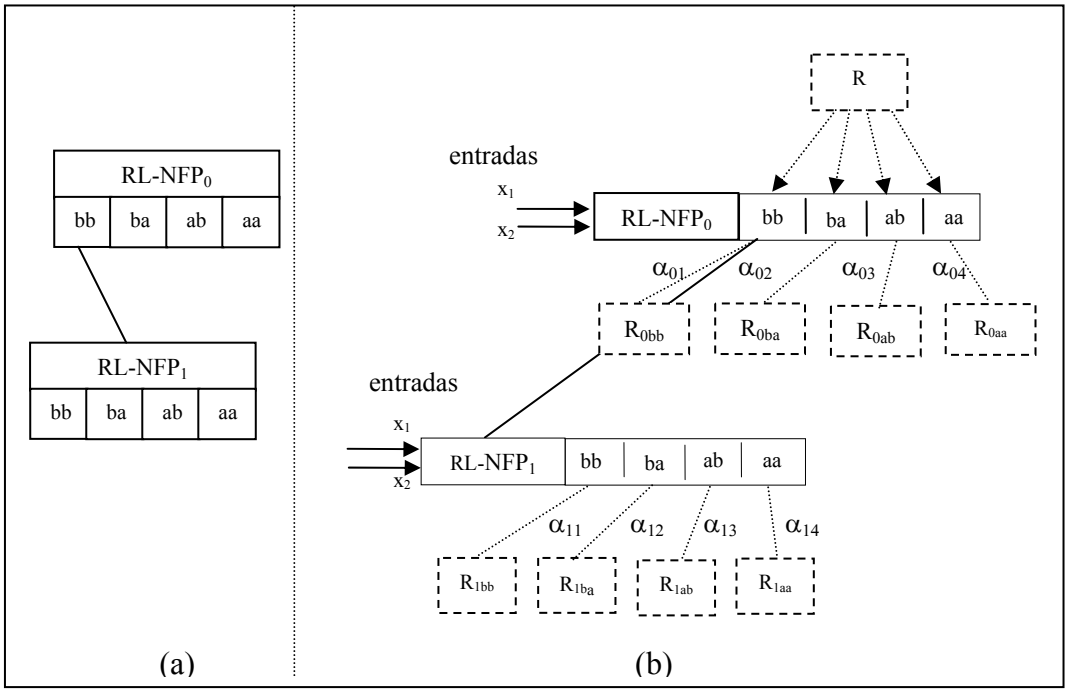

Figura 17: Retropropagação do retorno do ambiente para o modelo RL-NFHP.

Os valores α_i são calculados usando-se uma operação AND (T-norma). A figura 17 e a eq. 23 também exemplificam os cálculos dos α_i:

$$\begin{aligned} \alpha_{01} &= \rho_0(x_1) \,.\, \rho_0(x_2) \\ \alpha_{02} &= \rho_0(x_1) \,.\, \mu_0(x_2) \\ \alpha_{03} &= \mu_0(x_1) \,.\, \rho_0(x_2) \\ \alpha_{04} &= \mu_0(x_1) \,.\, \mu_0(x_2) \end{aligned} \tag{23}$$

Os cálculos dos retornos das partições da célula RL-NFHP$_0$ são definidos pela eq. 24.

$$\begin{aligned} R_{0bb} &= \alpha_{01} \,.\, R \\ R_{0ba} &= \alpha_{02} \,.\, R \\ R_{0ab} &= \alpha_{03} \,.\, R \\ R_{0aa} &= \alpha_{04} \,.\, R \end{aligned} \tag{24}$$

Os valores correspondentes aos retornos de cada partição da célula RL-NFP$_1$ são: R_{1bb}, R_{1ba} , R_{1ab} , R_{1aa} e são calculados mediante os graus de pertinências correspondentes aos α_i relativos à célula RL-NFP$_1$ onde:

$$\begin{aligned} \alpha_{11} &= \rho_1(x_1) \,.\, \rho_1(x_2) \\ \alpha_{12} &= \rho_1(x_1) \,.\, \mu_1(x_2) \\ \alpha_{13} &= \mu_1(x_1) \,.\, \rho_1(x_2) \\ \alpha_{14} &= \mu_1(x_1) \,.\, \mu_1(x_2) \end{aligned} \tag{25}$$

Os cálculos dos retornos da célula RL-NFHP$_1$ são definidos pelas eqs. 26.

$$\begin{aligned} R_{1bb} &= \alpha_{11} \,.\, R_{0bb} \\ R_{1ba} &= \alpha_{12} \,.\, R_{0bb} \\ R_{1ab} &= \alpha_{13} \,.\, R_{0bb} \\ R_{1aa} &= \alpha_{14} \,.\, R_{0bb} \end{aligned} \tag{26}$$

4 - Seleção das ações

As ações são associadas às funções de valores-Q e compõem um conjunto de ações que são selecionadas e experimentadas durante o aprendizado RL. Como já exposto na seções anteriores deste capítulo, a exploração do espaço de estados é fundamental para a descoberta de ações que correspondam à melhor resposta do agente (que visa atingir um objetivo) quando este se encontra em um determinado estado do ambiente. Sendo assim, para o modelo proposto, dois métodos de seleção foram testados.

O primeiro método, denominado ϵ-*greedy* (Sutton e Barto, 1998), seleciona a ação associada a maior função de valor-Q esperada com probabilidade $(1-\epsilon)$; e com probabilidade ϵ seleciona aleatoriamente uma ação qualquer.

No segundo método, a ação escolhida com probabilidade $(1-\epsilon)$ é também a que apresentar a maior função de valor-Q associada; e com um número de vezes proporcional a ϵ a seleção é baseada na distribuição de probabilidade dada pelas funções de valores-Q (eq. 27). Esta forma de seleção é mais conservativa, uma vez que as ações que apresentarem as maiores funções de valores-Q terão maiores chances de serem escolhidas. Este tipo de método de seleção de ações é conhecido como *softmax* (Sutton & Barto, 1998).

$$P(a_i|s) = \frac{Q(s,a_i)}{\sum_{a_k \in \{ações\}} Q(s,a_k)} \quad (27)$$

$P(a_i|s)$ é a probabilidade de se escolher a ação a_i, dado que o agente está no estado *s*, $Q(s,a_i)$ é a função de valor da ação a_i e $\sum_{a_k \in \{ações\}} Q(s,a_k)$ é a soma das funções de valores-Q relativas às ações disponíveis para o agente quando o mesmo se encontra no estado *s*.

O objetivo destes procedimentos é possibilitar a escolha de ações que resultem em um valor de reforço alto do ambiente, mas que ainda não tenham a função de valor-Q associada mais alta.

De uma forma geral, o segundo método de seleção de ações apresentado resultou em desempenho melhor do agente. Sendo assim, este foi o método adotado nos problemas definidos nos estudos de casos.

Idealmente, no início do processo de aprendizado, o parâmetro $\in$ deveria ter um valor que permitisse mais *exploration* (diversificação na escolha das ações) e menos *exploitation* (intensificação na escolha de determinadas ações) e, à medida que o sistema aprende, permitir menos *exploration* e mais *exploitation*. No entanto, como ocorrem inclusões de células na estrutura ao longo do processo de aprendizado, o que seria indicado, nestas circunstâncias, é que estas novas células também tenham oportunidade de explorar suas ações. Sendo assim, foi definido, para cada partição da célula, um parâmetro denominado $\in$*-greedy*, cujo valor deve estar em [0,1], possibilitando variar a política de escolha da ação deste modelo.

Com o objetivo de melhorar o desempenho do aprendizado do modelo, o parâmetro $\in$*-greedy* foi definido de forma *adaptativa.* Este procedimento é usado no método de aprendizado por reforço AHC (Sutton 1998). Neste procedimento, quando a ação resultante é boa, o valor do parâmetro $\in$*-greedy* desta partição é reduzido, diminuindo a probabilidade desta partição usar uma política *explorative* (seleção da ação associada a maior função de valor Q). Quando a ação resultante não apresenta um bom desempenho, as partições das células ativas têm os seus parâmetros $\in$*-greedy* aumentados, permitindo que estas partições, nos passos

seguintes, tenham maiores chances de usarem um método *exploitive* (seleção da ação aleatória da ação).

5 - Atualização dos valores Q

A partir dos valores de reforços calculados para cada célula da estrutura, as funções de valores-Q associadas às ações que tenham contribuído para a ação resultante executada pelo agente devem ser atualizadas.

Esta atualização é feita a partir da avaliação entre os retornos globais atual e anterior. A atualização das funções de valores-Q ocorre de duas formas distintas: para o caso do valor do retorno global atual ser maior que o retorno global anterior ($R_{t+1} > R_t$) e para o caso do retorno global atual ser menor ou igual ao retorno global anterior ($R_t \geq R_{t+1}$).

Primeiro Caso: $R_{t+1} > R_t$

Caso o retorno global atual seja maior que o retorno global anterior, então as ações atuais (ações que foram executadas quando o agente estava no estado s_t) têm maiores chances de serem a melhor resposta do agente ao sistema quando o mesmo se encontrar neste estado. Assim, se $R_{t+1} > R_t$ deve-se premiar as ações selecionadas no passo (t), atualizando suas respectivas funções de valores-Q conforme a equação do algoritmo *SARSA* (Sutton, 1996), definida pela seguinte expressão:

$$Q(s_t, a_t) = (1 - \alpha_t) \, . \, Q(s_t, a_t) + \alpha_t \, [\, r_{t+1} + \gamma \, Q(s_{t+1,} \, a_{t+1})] \qquad (28)$$

primeira parcela segunda parcela

onde: o valor $Q(s_t, a_t)$ é atualizado a partir do seu valor atual; r_{t+1} é o reforço local imediato (este é o reforço que é definido no passo da retropropagação do retorno global); o γ é um parâmetro que fixa um percentual da contribuição da função de valor-Q associada à próxima ação a_{t+1} escolhida ($Q(s_{t+1,} \, a_{t+1,})$) quando o sistema está no estado s_{t+1}; e α é o parâmetro proporcional à contribuição

relativa desta ação local na ação global. A seguir o parâmetro α e a definição do valor $Q(s_{t+1}, a_{t+1},)$ usados na eq. 28 serão detalhados.

O parâmetro α

O parâmetro α está compreendido entre [0,1]. Na maioria das aplicações ele tem seu valor inicial igual a 1 e, à medida que o aprendizado evolui, seu valor é reduzido. No início do processo de aprendizado, sua função é estimular os novos valores aprendidos a partir do reforço (r_{t+1}) e de $Q(s_{t+1}, a_{t+1})$ (Sutton, 1998). À medida que o processo de aprendizado evolui, o valor de α é reduzido, aumentando o peso da primeira parcela, que é relativa aos valores $Q(s_t, a_t)$ já aprendidos.

No caso do modelo RL-NFHP (Figueiredo, 2003), baseado em aproximação de funções, a redução deste parâmetro ao longo do processo resultou em graves problemas de aprendizado. A explicação para este fato é que quando os graus de pertinência das bipartições são pequenos (podendo permanecer assim durante um número significante de passos), a contribuição da segunda parcela da eq. 28 não era tão efetiva e o valor de α ainda alto não permitia que a função de valor-Q pudesse evoluir, ou seja, aumentar e se destacar dos demais. Quando o grau de pertinência desta bipartição aumentava, muitas vezes o parâmetro α tinha o seu valor já muito reduzido, não permitindo que a atualização da função de valor-Q desta partição fosse realizada adequadamente, nem para este valor Q, nem para outro que pudesse vir a ser escolhido por qualquer método *non-greedy*.

Após vários testes, a avaliação que gerou melhores resultados foi a que definiu o parâmetro α como sendo proporcional à contribuição relativa desta ação local na ação global. Como a punição e o prêmio são realizadas em condições distintas, a atualização da função de valor-Q da partição que estiver contribuindo mais naquele passo também terá seu valor atualizado segundo esta proporção e a que tiver participação minoritária terá seu valor alterado na proporção desta participação.

A ação de saída da célula é resultado da contribuição das ações de seus dois conseqüentes. Caso a bipartição que está sendo atualizada tenha um grau de pertinência muito pequeno, mesmo que a ação não seja uma ação ideal, essa

influência é minimizada. À medida que o agente se desloca no espaço de estados e cresce o grau de pertinência desta partição, a ação "não ideal" que antes tinha seu peso reduzido graças ao grau de pertinência menor, passa a acarretar uma saída "ruim". Por isso a atualização da função de valor-Q (no que diz respeito ao retorno e ao valor de $Q(s_{t+1}, a_{t+1})$) deve depender do grau de importância que esta ação tem na saída.

Em outras aplicações, nas quais as modelagens diferem da modelagem tradicional de RL (como a *lookup table*), os autores também ajustaram este parâmetro segundo as necessidades de seus modelos e obtiveram bons resultados. Sutton (1998), para a aplicação do carro da montanha usando o modelo CMAC, define o α como uma constante. Jouffe (1998); também utilizou uma forma adaptativa para α, na qual o parâmetro pode crescer ou decrescer segundo sua heurística definida para o aprendizado.

O valor $Q(s_{t+1}, a_{t+1})$

Após a execução da ação resultante, o agente passa ao estado s_{t+1} do ambiente. Isso significa que pelo menos duas funções de valores-Q (correspondentes às ações selecionadas a_{t+1} para as bipartições baixo e alto) devem ser consideradas para $Q(s_{t+1}, a_{t+1})$ da equação 28.

A figura 18 exemplifica esta situação. No estado s_t a célula ativa apresenta duas funções de valor a serem atualizadas: Q_1 no conjunto de ações relativo à bipartição baixo e Q_2 no conjunto de ações relativo a bipartição alto. Quando o agente passa ao estado seguinte, s_{t+1}, após a execução da ação resultante, a célula que é ativada também seleciona duas ações (a_{t+1}), cada uma associada a sua função de valor-Q (neste caso, Q_2 e Q_3). Para a atualização dos valores de Q_1 e Q_2 relativos ao estado s_t, duas propostas foram testadas:

- na primeira proposta o valor $Q(s_{t+1}, a_{t+1})$ do estado s_{t+1} considerado na atualização é aquele que corresponder ao ramo da estrutura que apresentar maior peso na ação de saída; neste caso seria o valor de Q_2, se $\alpha_1 > \alpha_2$, ou Q_3, se $\alpha_2 > \alpha_1$;

- na segunda, o valor $Q(s_{t+1}, a_{t+1})$ é calculado a partir da soma ponderada de Q_2 e Q_3 com relação aos graus de pertinência da variável de entrada da célula $(\alpha_1 Q_2 + \alpha_2 Q_3)$.

Apesar dos resultados não diferirem significativamente, o segundo método foi o adotado por apresentar resultados, em média, ligeiramente superiores ao primeiro.

O valor-$Q(s_{t+1}, a_{t+1})$ da ação escolhida para o estado s_{t+1}, que será usado para atualizar as funções de valores-Q (relativos aos conjuntos baixo e alto) quando o agente está no estado s_t , também considera o peso que cada ação escolhida (a_t) no estado s_t teve na ação resultante.

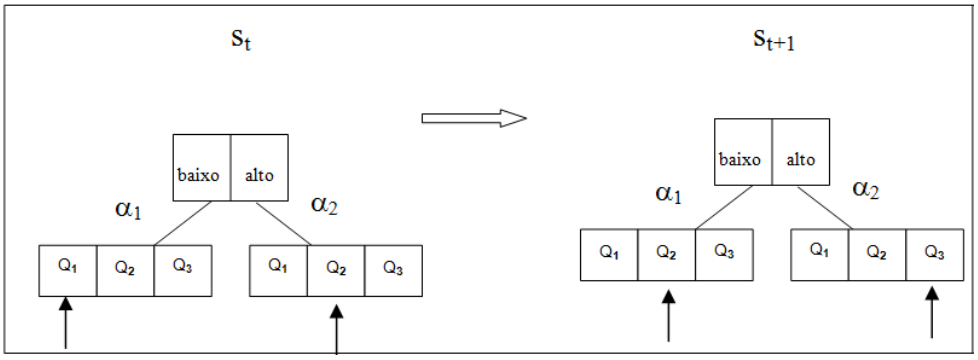

Figura 18: Caso exemplo de atualização da função de valor Q_{t+1} relativa à equação 19.

Neste passo do algoritmo também é atualizada a taxa de *exploration/exploitation* já mencionada anteriormente, ϵ-greedy. Cada partição deve ter o seu parâmetro ϵ-*greedy* reduzido/aumentado segundo um pequeno percentual (por exemplo, 5% do seu valor). Este procedimento segue as recomendações feitas por Sutton (1996). O parâmetro ϵ-*greedy* não deve ser superior a 10%.

Segundo Caso: $R_t \geq R_{t+1}$

Caso o reforço global atual seja menor ou igual ao reforço global anterior, isto significa que as ações atuais não maximizam a função de reforço do estado em que o agente se encontra. Sendo assim, essas ações devem se tornar menos

propensas a serem selecionadas nas próximas vezes em que as células, às quais elas pertencem, estiverem ativas.

Se $R_t \geq R_{t+1}$, as ações são punidas, reduzindo suas funções de valores-Q na proporção da contribuição do reforço local desta bipartição da célula e o reforço global. A explicação para esta medida é a mesma já descrita antes. Caso a influência de uma ação boa seja minimizada pelo seu grau de pertinência neste momento, não se deseja que sua função de valor-Q seja drasticamente reduzida, devido à má influência da ação da bipartição irmã. A função de valor-Q é atualizada como na equação 29.

$$Q(a_t, s_t) = (1 - fp) \, . \, Q(a_t, s_t) \qquad (29)$$

Na equação acima, fp é o fator de punição, que varia entre [0,1] e é definido como a relação entre o retorno local da partição e o retorno global.

Caso as ações escolhidas sejam mal sucedidas para um determinado estado, os parâmetros ϵ-*greedy* das partições envolvidas terão suas taxas aumentadas, permitindo que, nas próximas vezes que esta partição estiver ativa, outras ações diferentes tenham mais chances de serem escolhidas. Isso se aplica a qualquer dos métodos de seleção descritos.

Desta forma, é feito o aprendizado das ações que serão executadas quando o agente se encontra em um determinado estado. Este estado é definido pelas células que estão ativas a cada passo.

6 - Particionamento das células

Com relação ao crescimento da estrutura, algumas avaliações tornam-se necessárias para se garantir o aprendizado. Para isso, foram criados o parâmetro **variável de crescimento** e a **função de crescimento**. Essa variável e esta função têm como objetivo permitir ou limitar o crescimento da estrutura. À medida que as células são atualizadas, verifica-se o percentual de variação da função de valor Q (ΔQ) das ações associadas às partições baixas e altas das células. Quando a variação da função de valor-Q associada à ação atualizada é maior que um percentual da maior variação já ocorrida para esta partição da célula, considera-se que esta partição apresenta potencial de crescimento, ou seja, esta variação pode

indicar que as ações que estão sendo tomadas nesta bipartição não estão adequadas para o sub-domínio relativo a esta bipartição. Logo:

Se $\Delta Q > p * \Delta\vartheta$, então a **variável de crescimento** para esta bipartição da célula é incrementada. ΔQ é a variação percentual da função de valor-Q neste passo; $p \in [0,1]$ é um percentual que representa o percentual de atualização da função de valor-Q em relação a maior variação já ocorrida até o momento; $\Delta\vartheta$ registra a maior variação da função de valor-Q da partição de uma célula ao longo do aprendizado a cada ciclo.

Se $\Delta Q < p * \Delta\vartheta$, então a **variável de crescimento** para esta bipartição da célula é reduzida.

A definição do percentual p está associada diretamente à taxa de crescimento desta estrutura: quanto maior for o valor de p, maior variação de ΔQ será permitida nesta bipartição.

O objetivo deste percentual é permitir que haja alguma variação na atualização da função de valor-Q desta bipartição, mas sem prejudicar o aprendizado. Pequenas variações ocorrem principalmente no início do aprendizado (já que a função de valor-Q inicial é zero) ou quando o processo de exploração escolhe uma ação diferente (com sua função de valor-Q associada) que passa a maximizar o valor de retorno para aquele estado. No entanto, variações muito grandes indicam que a ação tomada nesta bipartição não responde adequadamente ao comportamento desejado (definido através do retorno do ambiente) às exigências deste estado (domínio), ou seja, esta ação não consegue atender ao comportamento desejado para o agente em todo este domínio, indicando que ele deve ser particionado.

Associada à variável critério de aprendizado deve-se definir a função de crescimento, cujo objetivo é limitar o crescimento ao longo do processo de aprendizado. Idealmente, ela deve ser menos exigente com relação às células iniciais do sistema, ou seja, deve permitir que no início do aprendizado as células se multipliquem mais rapidamente e, à medida que o sistema evolui, deve crescer o grau de exigência da função, ou seja, aumentar o efeito de *exploration/exploitation* sobre as ações das células da estrutura. Isso se deve ao fato de que as ações das células criadas no início do aprendizado não são tão efetivas para domínios ainda muito abrangentes.

A cada ciclo avalia-se se a variável de crescimento da bipartição de uma célula tornou-se maior do que a **função de crescimento**. Caso isso seja verdade, então esta bipartição apresenta a primeira condição para gerar uma célula filha.

Esta função de crescimento, definida heuristicamente, pode ser uma constante (como em (Pyatt & Howe, 1998)) ou ser função do número de passos, tamanho da estrutura (profundidade da árvore), ou qualquer outra função relacionada ao processo ou ao objetivo do aprendizado. Nos testes apresentados neste trabalho a função de crescimento é função do número de passos e do número de ciclos, como mostra a figura 19. O eixo x representa os ciclos e o eixo y representa o valor limite para variável de crescimento.

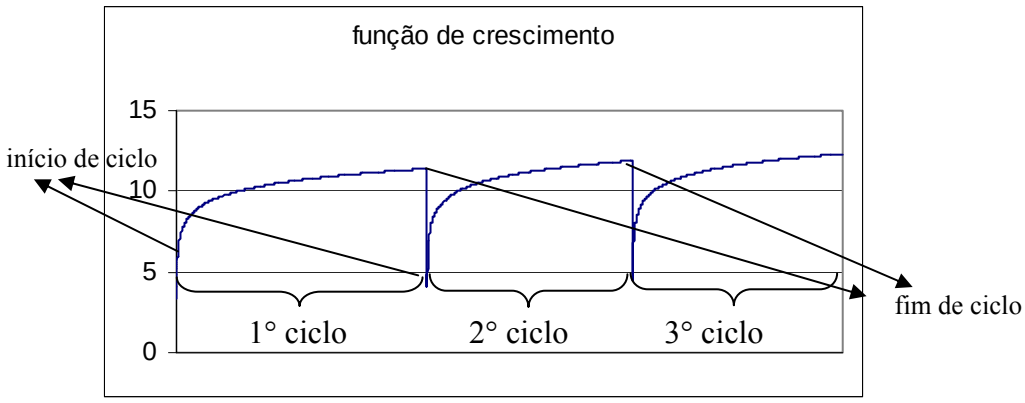

Figura 19: Função de crescimento: (log(n x número de passos x número de ciclos), onde n > 1.

Ao longo dos ciclos, a função de crescimento aumenta o valor que a variável critério de aprendizado deve atingir para que ocorra um novo particionamento. No entanto, a cada novo ciclo, a função de crescimento apresenta valor maior do que o valor do início do ciclo anterior e menor do que o valor do final do ciclo deste ciclo. Isso permite que as novas células criadas a cada ciclo também tenham chances de ser particionadas.

O valor ΔQ é armazenado em uma lista, a qual será posteriormente avaliada em relação à dispersão destes valores. Este é o segundo critério adotado usado em juntamente com o primeiro critério, para se determinar se deve ou não ocorrer o particionamento de uma célula. Então, o crescimento da estrutura acontece segundo estes dois critérios adotados:

1 – o valor do critério de aprendizado desta partição for maior do que a função de crescimento;

2 – $|\mu| < 2\,\sigma$, onde μ e σ são a média e o desvio padrão de ΔQ.

Caso o aprendizado das ações de cada uma das partições das células não seja efetivo, ou seja, caso a variável de crescimento de qualquer uma das partições (ou mesmo das duas) da célula atinja o valor definido pela função de crescimento, a partição apresenta o primeiro requisito para o particionamento.

O segundo requisito para o particionamento é que o valor da média da variação da função de valor-Q desta partição seja menor em módulo do que duas vezes o seu desvio padrão.

Estes mecanismos também foram usados em outros modelos por outros pesquisadores em sistemas de aprendizado baseados em árvores (Pyatt & Howe, 1998; Uther & Veloso, 1998).

Sendo assim, quanto mais rapidamente a bipartição baixa e/ou alta da célula ultrapassar os limites de capacidade de aprendizado (por não conseguir aprender adequadamente), mais rapidamente ela será particionada, de forma a especializar o domínio da célula que não conseguiu atingir seus objetivos.

A função de crescimento apresenta um valor pequeno no início do aprendizado (reduzindo o grau de exigência para realizar o particionamento), quando os conjuntos fuzzy das células ainda são abrangentes, com relação ao domínio, e maior ao final do processo (aumentando o grau de exigência para realizar o particionamento), quando os conjuntos fuzzy já se especializaram o suficiente para garantir o aprendizado.

Quando uma polipartição possuir todos os requisitos necessários para o particionamento, uma célula filha é criada e conectada àquela polipartição. Seu domínio será o subdomínio correspondente à polipartição do seu ancestral. As células filhas também herdam da partição ancestral o conjunto de ações com seus respectivos valores Q.

Na figura 20, a célula raiz (ou célula pai) possui os domínios definidos pelos intervalos (x_{1LI} , x_{1LS}) para a entrada x_1 e (x_{2LI} , x_{2LS}) para a entrada x_2. A célula filha 1 descende da polipartição referente à composição do conjunto baixo relativo à entrada x_1 e ao conjunto baixo da entrada x_2 da célula raiz (ou célula

pai). Seus domínios são, portanto, diretamente relacionados aos subdomínios da partição baixo/baixo da célula pai (RL-NFHP$_0$) e são definidos por (x_{1LI}, $x_{1(LS+LI)/2)}$) relativo à entrada x_1 e por (x_{2LI}, $x_{2(LS+LI)/2)}$) relativo à entrada x_2.

Quando uma célula (raiz ou pai) possuir todas as células descendentes (no caso exemplo da figura 20 quatro células), todas as ações da célula pai tornam-se as ações de saída das células filhas e neste caso têm suas funções de pertinência anuladas. Ou seja, os graus de pertinência ρ e μ da célula pai passam a valer 1.

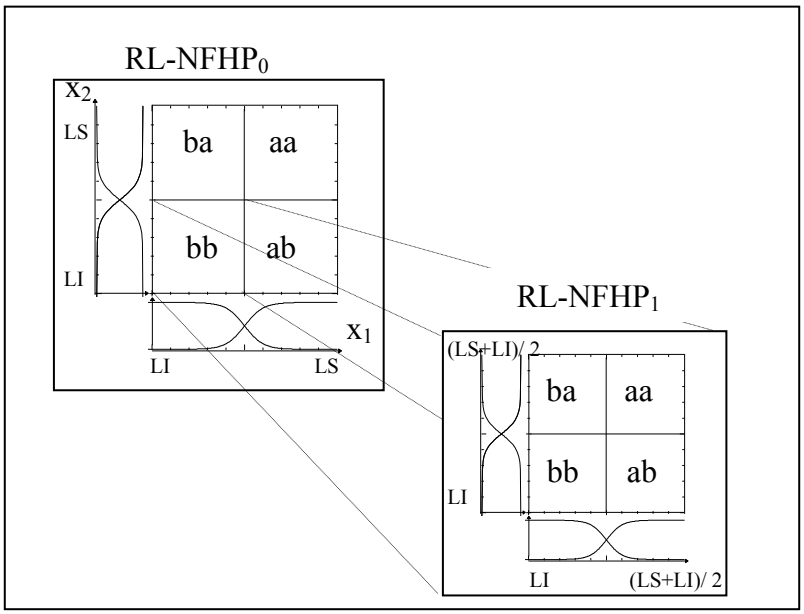

Figura 20: Particionamento da Célula RL-NFP.

Como já mencionado no modelo descrito no capítulo anterior, este procedimento contorna o problema de se ter na saída da estrutura valores de ações muito pequenos e não prejudica o mapeamento (estado-ação), uma vez que o domínio da célula pai já está representado nas células filhas com um grau de precisão maior (ver seção 3.5).

O próximo capítulo apresentará a versão multi-agentes dos algoritmos da família RL-NFH. A principal motivação para o desenvolvimento destes novos modelos é fazer com que cada agente possa explorar diferentes tarefas simultaneamente, acelerando o aprendizado e a convergência para uma política ótima. O capítulo 4 também apresentará os princípios de *satisfatoriedade* e *não-dominação* (Goodrich e Quigley, 2004), que serão incorporados ao modelo RL-NFHP (Figueiredo, 2003), com o objetivo de superar algumas limitações existentes.