Aprendizado por Reforço para um Sistema Tutor Inteligente sem Modelo Explícito do Aprendiz

Marcus Vinícius Carvalho Guelpeli
Divisão de Ciência da Computação
Instituto Tecnológico de Aeronáutica
São José dos- Campos São Paulo-SP

guelpeli@terra.com.br

Nizam Omar
Depto. de Engenharia Elétrica
Universidade Presbiteriana Mackenzie, São Paulo-SP

omar@comp.ita.br

Carlos Henrique Costa Ribeiro
Divisão de Ciência da Computação
Instituto Tecnológico de Aeronáutica
São José dos- Campos São Paulo-SP

carlos@comp.ita.br

Resumo *Este trabalho tem como meta apresentar um módulo de diagnóstico incluído na arquitetura tradicional de Sistemas Tutores Inteligentes, no qual é aplicada uma técnica de Aprendizado por Reforço (algoritmo Q-Learning) que possibilita a modelagem autônoma do aprendiz. Um valor de utilidade é calculado baseado em uma tabela de pares estado-ação, a partir da qual o algoritmo estima recompensas futuras que representam os estados cognitivos do aprendiz. A melhor política a ser usada pelo tutor para qualquer estado cognitivo é então aprendida e disponibilizada pelo algoritmo de Aprendizagem por Reforço.*

Palavras-chave: Sistemas Tutores Inteligentes, Aprendizado por Reforço, Algoritmo Q-Learning.

Abstract *The goal of this paper is to present a diagnostic module included in an Intelligent Tutoring System (ITS) architeture. In this module, a Reinforcement Learning technique (Q-Learning algorithm) is applied, making it possible to autonomous modelling of the learner. An utility value is calculated based on a state-action table upon which the algorithm estimates future rewards which represent the cognitive states of the learner. The best action policy to be used by the tutor at any cognitive state is then learned and made available by the Reinforcement Learning algorithm.*

Keywords: *Intelligent Tutoring Systems, Reinforcement learning, Q-Learning algorithm*

1 Introdução

Um Sistema Tutor Inteligente (STI) é uma evolução de sistemas CAI (*Computer-Assisted Instruction*), aperfeiçoado com técnicas de Inteligência Artificial (IA). Possibilitam ao estudante a capacidade de aprender com um tutor artificial, que serve como guia no processo. Este último deve se adaptar ao aprendiz, e não o contrário, como acontece em métodos tradicionais. Com isso, é necessário um modelamento do aprendiz, para que o STI possa saber o que ensinar, a quem ensinar e como ensinar. O STI deve ser capaz de mudar o nível de entendimento para responder às entradas do aprendiz, podendo mudar as estratégias pedagógicas de forma individualizada e de acordo com o ritmo e as características de cada aprendiz [3].

Para ser inteligente, um tutor deve ser flexível, isto é, ter capacidade para aprender com o meio ambiente e atualizar seu conhecimento [8]. Este artigo propõe um processo de definição de perfil e o uso de um algoritmo de Aprendizado por Reforço (algoritmo Q-learning), através da introdução de um módulo de diagnóstico em uma arquitetura de STI, com o propósito de incorporar ao tutor a capacidade de modelar autonomamente o aprendiz, através de interações com o seu ambiente de atuação.

O trabalho está organizado da seguinte forma. Na Seção 2 apresentam-se os conceitos de Aprendizado por Reforço e o algoritmo Q-Learning. A Seção 3 contém uma descrição da estrutura do STI utilizando AR. A Se-

ção 4 contém experimentos e resultados. Finalmente, a Seção 5 apresenta as vantagens e desvantagens do método proposto, e sugere trabalhos futuros.

2. Aprendizado por Reforço

A arquitetura descrita neste trabalho utiliza uma técnica de Inteligência Artificial conhecida como Aprendizado por Reforço (AR), para modelar o aprendiz de forma dinâmica e autônoma. AR permite ao agente adquirir uma capacidade de conhecimento do ambiente que não estava disponível em tempo de projeto [6]. É baseado na existência de um crítico que apenas avalia a ação tomada pelo agente aprendiz , sem indicar explicitamente a ação correta. Formalmente, AR utiliza uma estrutura composta de estados, ações e recompensas conforme mostra a Figura1.

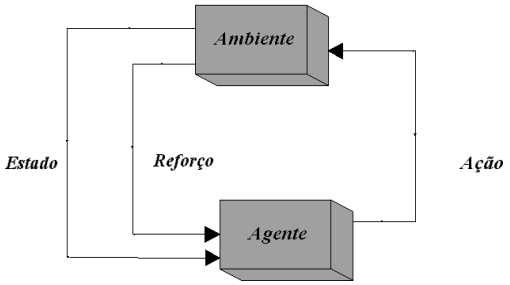


Figura 1. Um agente AR interagindo com seu ambiente.

O agente atua em um ambiente descrito por um conjunto de possíveis estados e pode executar, para cada estado, uma ação dentro de um conjunto de ações possíveis, recebendo um valor de reforço ou recompensa cada vez que executa uma ação. Este reforço indica o valor imediato da transição estado $\Rightarrow$ ação $\Rightarrow$ novo estado. Ao longo do tempo, este processo produz uma seqüência de pares estado-ação, e seus respectivos valores de reforços. O objetivo do agente é aprender uma política μ que maximize o valor esperado da soma destes reforços a longo prazo.

Uma consideração importante em AR refere-se ao conflito exploração X explotação [1]. O agente deve lidar com um compromisso entre escolher a exploração de estados e ações desconhecidos, de modo a coletar mais informação, ou a explotação de estados e ações que já foram aprendidos e que irão gerar altas recompensas, de modo a maximizar seus reforços acumulados. Assim, por um lado o agente deve aprender quais ações maximizam os valores das recompensas, e, por outro deve agir de forma a atingir esta maximização. É importante, portanto, estabelecer uma política mista de explotação e exploração (política ε-*greedy*), que inclua a escolha intencional (com probabilidade ε) de se executar uma ação aleatória não-ótima (considerando-se o estado de conhecimento atual), visando a aquisição de conhecimentos a respeito de estados ainda desconhecidos ou pouco visitados. Em uma política de explotação pura (*greedy*) escolhem-se as ações julgadas (talvez erroneamente, caso o algoritmo de AR ainda esteja longe da convergência) serem as melhores para maximixar a soma esperada das recompensas.

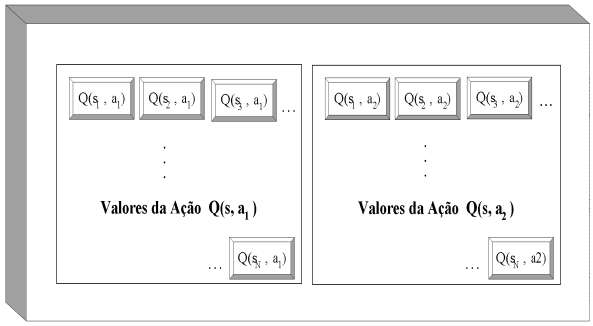
No contexto deste trabalho, a técnica de AR terá como função modificar parâmetros e armazená-los em esquemas de planos, que definem a forma de apresentar o material instrucional ao aprendiz. O desafio, neste caso, é escolher a melhor ação que – baseada no estado do aprendiz – possa mudar a estratégia de ensino, para obter resultados significativos, fornecedores de parâmetros indicativos de quão boa ou ruim está sendo uma determinada estratégia. O algoritmo utilizado (Q-learning) é baseado em estimativas de utilidades de pares estado-ação, e com estas estimativas podem-se gerar alternativas de novas estratégias pedagógicas. As estimativas destas utilidades – baseadas nas quais será portanto feita a aprendizagem das estratégias mais adequadas, que indicam a ação a escolher para cada estado do aprendiz – dar-se-á através de uma estrutura de parâmetros adaptativos sobre os quais o algoritmo opera.

Formalmente, AR procura aproximar uma função que define a utilidade relativa dos pares estado-ação. Esta função de utilidade fornece indicações estimativas, mapeando os pares estado-ação em uma medida baseada na soma dos reforços esperados a longo prazo. Para cada par estado-ação (s,a), é definido o reforço r(s,a), indicando uma conseqüência *imediata* da execução da ação a no estado s. O problema em AR é achar uma política ótima de ações $\mu^*(s)$, ou seja, um conjunto de ações que maximizem, para cada estado s, os valores de utilidade Q(s,a).

2.1 Q-Learning

O algoritmo Q-Learning [9], pode ser usado para definir a escolha da melhor ação em AR. Neste algoritmo, a escolha de uma ação é baseada em uma função de utilidade que mapeia estados e ações a um valor numérico. No trabalho descrito neste artigo, o valor de utilidade Q(s,a) de um par (s,a), é calculado a partir de reforços medidos pela qualidade do estado cognitivo do aprendiz (módulo de diagnóstico). Cada valor Q(s,a) representa o valor esperado da soma de reforços ao se executar a ação a no estado s, seguindo-se uma política ótima a partir de então. Portanto, uma vez que os valores Q(s,a) estejam bem estimados, a melhor ação a ser executada no estado s pode ser obtida simplesmente como $\arg\max_a Q(s,a)$. O principal objetivo do algoritmo Q-Learning é portanto

estimar o valor Q(s,a) para cada possível ação a e estado s, e a partir daí permitir a obtenção da melhor ação (ou seja, a ação com maior valor de utilidade).



Q-Learning normalmente usa uma tabela (Figura 2) para armazenar os valores de utilidade Q(s,a) estimados para os pares (estado, ação).

Figura 2. Tabela dos pares (estado - ação).

A atualização dos valores Q é feita da seguinte maneira:

Inicialize Q(s,a).
Para cada instante t repita:

1. Observe estado $\mathbf{s_t}$ e escolha uma ação $\mathbf{a_t}$;
2. Observe o estado $\mathbf{s_{t+1}}$ e atualize $\mathbf{Q_t(s_t, a_t)}$ de acordo com: $\mathbf{Q_{t+1}(s_t,a_t) = Q_t(s_t,a_t) + \alpha_t[r(s_t) + \gamma max_a Q_t(s_{t+1},a) - Q_t(s_t,a_t)];}$
   até **t** igual a limite de passos.

Onde:

- t=0,1,2,3,....
- $Q_{t+1}(s_t,a_t)$ é o valor (qualidade) da ação a_t no estado s_t, estimado no instante t+1.
- $r(s_t)$ é o reforço imediato recebido no estado s_t.
- $0 < \alpha_t \leq 1$ é a taxa de aprendizado.
- γ é uma taxa de desconto temporal. Quanto mais próximo de 1 for o valor de γ, maior importância é dada aos reforços mais distantes no tempo.
- $max_a\ Q_t(s_{t+1},a)$ é o valor Q correspondente à ação com maior valor de utilidade em s_{t+1}.

No início, o algoritmo vai estar longe da utilidade ótima associada ao estado s, mas com o passar do tempo as estimativas de Q melhoram. De fato, os valores Q atualizados pelo algoritmo Q-Learning convergem para os corretos, desde que os pares (s,a) tenham sido visitados um número infinito de vezes [10]. Na prática, convergência para políticas de ação de boa qualidade é obtida com exploração adequada do espaço de estados, durante um número razoável de iterações. A política de ações durante a execução do algoritmo nas fases iniciais de treinamento deve portanto garantir uma boa exploração do espaço de estados (política ε-greedy).

3 Estrutura do Sistema

A Figura 3 ilustra a estrutura proposta neste artigo para o funcionamento da Técnica de Aprendizado por Reforço em um STI.

A complexidade de estabelecer um perfil está na modelagem do aprendiz. O perfil vai indicar o estado cognitivo do mesmo a partir do comportamento observável e, com isso, poder indicar uma estratégia de ensino a ser utilizada. Decisões pedagógicas são um desafio para os tutores, já que as pessoas nem sempre conseguem representar seus próprios processos mentais. Segundo [2] um modelo realista do aluno implica uma atualização dinâmica enquanto o sistema avalia o seu desempenho. Este modelamento inicialmente dá-se através de questionários, a partir do qual é conhecido o perfil do aprendiz. Logo em seguida o sistema gera um ciclo que classifica o aprendiz dentro de faixas distintas de conhecimentos, de forma dinâmica e usando as informações do módulo diagnóstico.

É conveniente manter um log de respostas dadas, para que seja desnecessária a repetição dos questionários e seja possibilitada a continuidade do processo.

O módulo diagnóstico produz – de acordo com a política de ações estabelecida – uma indicação de quão bom está sendo o desempenho do aprendiz em relação às ações produzidas pelo tutor. No contexto de AR, o módulo de diagnóstico envia reforços indicando o desempenho, produzindo para cada par (s,a), um valor r(s,a), correspondendo à reforços positivos para resultados favoráveis e reforços negativos para resultados desfavoráveis. Os valores $Q_t(s,a)$ são então atualizados na tabela Q (Figura 2), para que, com base nesses valores, o tutor possa reclassificar o aprendiz.

O sistema classifica o estado cognitivo do aprendiz através do resultado do questionário, usando uma escala de classificação (Figura 4) e obtendo seu perfil. Com o perfil (estado) definido, o sistema escolhe a ação com o maior valor de utilidade, baseado nos valores Q(s,a) armazenados na tabela (Figura 2). As ações representam uma estratégia pedagógica adotada pelo tutor para guiar o aprendiz.

Os resultados são passados ao Módulo Diagnóstico, que calcula os reforços para o par (s,a) e atualiza o valor $Q_t(s,a)$, usando o algoritmo Q-learning. Após atualizado o valor de utilidade, o sistema volta a reclassificar o aprendiz na escala.

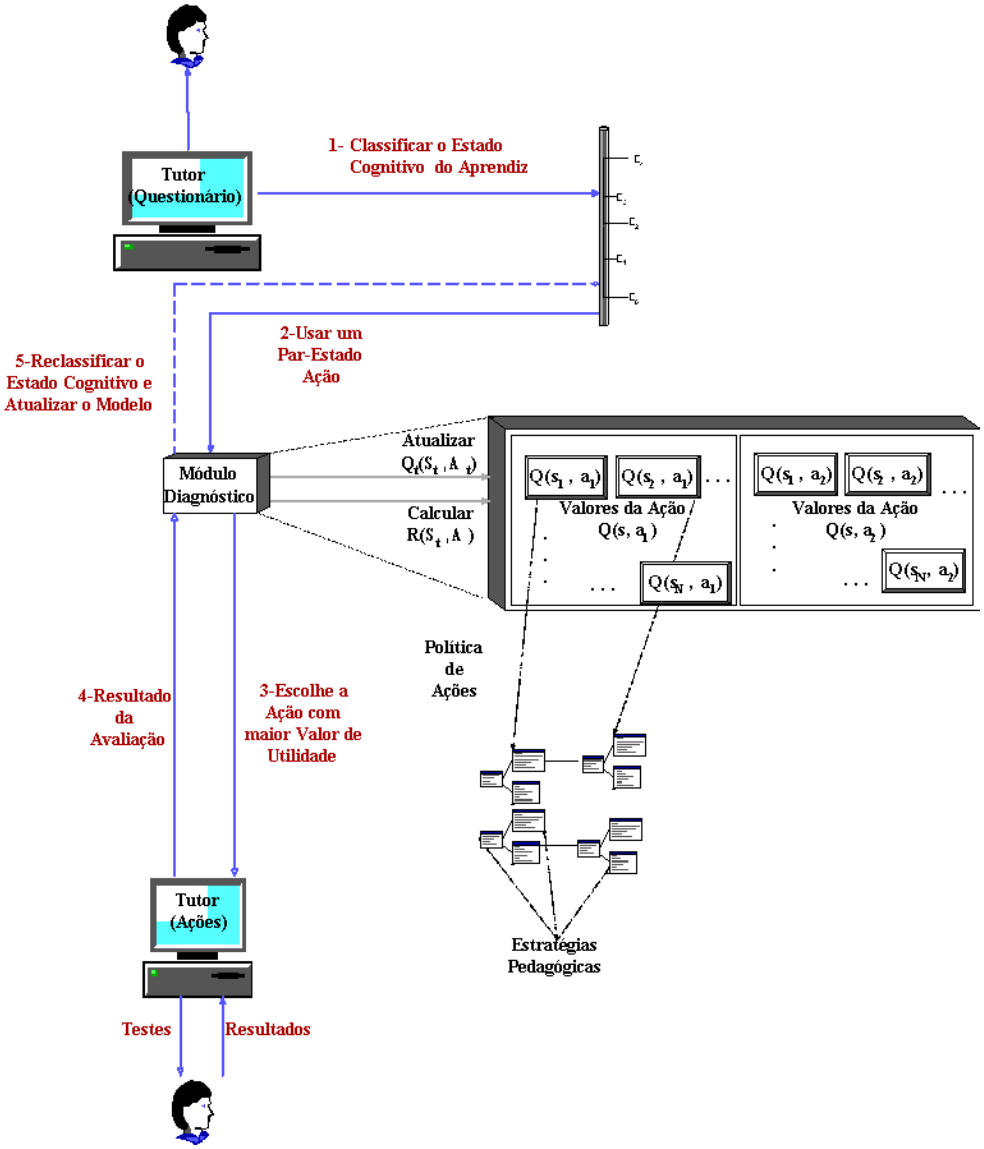


Figura 3. STI com a Técnica de AR.

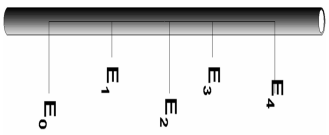


Figura 4. Classificação do Estado Cognitivo do Aprendiz.

4 Experimentos

Os experimentos foram realizados sobre um protótipo de STI, desenvolvido em linguagem C. Uma matriz 5x10 mapeando pares estado-ação (5 estados e 10 ações) em valores Q foi definida com os elementos descritos a seguir.

- Um conjunto de estados S={E_0, E_1, E_2, E_3, E_4}, onde cada um representa um possível estado cognitivo do aprendiz, em face da interação com o tutor, ou seja, é o resultado obtido pelo tutor ao aplicar uma ação A_i em um dado instante de tempo. Cada estado corresponde a um grau de evolução do aprendiz. Assim, define-se o mapeamento:

$$E_0 \Rightarrow [0,2]$$
$$E_1 \Rightarrow]2,4]$$
$$E_2 \Rightarrow]4,6]$$
$$E_3 \Rightarrow]6,8]$$
$$E_4 \Rightarrow]8,10]$$

Os valores de 0 a 10, divididos nos cinco conjuntos em cada estado representam uma métrica usada para diferenciar os estados.

- Um conjunto de ações A={A0, ..., A9} que podem ser escolhidas pelo tutor. Cada ação pode corresponder à aplicação de provas, exercícios, questionários, perguntas, trabalhos, testes, etc, ou combinações destes e outros dispositivos de avaliação usados pelo tutor, segundo as estratégias pedagógicas estabelecidas.

- Um conjunto de reforços instantâneos associados a cada estado visitado, definidos da seguinte forma: $E_0 \Rightarrow$ R=1 (Ruim); $E_1 \Rightarrow$ R=3 (Regular); $E_2 \Rightarrow$ R=5 (Bom); $E_3 \Rightarrow$ R=7 (Muito Bom); $E_4 \Rightarrow$ R=10 (Excelente)

Foi definida uma metodologia de teste na qual foram criados três modelos não-determinísticos:

a) Modelo M1 (Ruim), definido por um conjunto de estados e ações de acordo com as tabelas 1,2,3,4 e 5 de transição de estados $P(s_t \mid s_{t-1}, a)$.

b) Modelo M2 (Bom), definido por um conjunto de estados e ações de acordo com as tabelas 6,7,8,9,10 de transição de estados $P(s_t \mid s_{t-1}, a)$.

c) Modelo M3 (Excelente), definido por um conjunto de estados e ações de acordo com as tabelas 11,12,13,14 e 15 de transição de estados $P(s_t \mid s_{t-1}, a)$.

TABELA 1: $P(s|E_0,a)$, Modelo M_1.

	E_0	E_1	E_2	E_3	E_4
A_0	0.8	0.15	0.05	0	0
A_1	0.8	0.15	0.05	0	0
A_2	0.8	0.15	0.05	0	0
A_3	0.8	0.15	0.05	0	0
A_4	0.8	0.15	0.05	0	0
A_5	0.8	0.15	0.05	0	0
A_6	0.8	0.10	0.05	0.03	0.02
A_7	0.8	0.10	0.05	0.03	0.02
A_8	0.8	0.10	0.05	0.03	0.02
A_9	0.8	0.10	0.05	0.03	0.02

TABELA 2: $P(s|E_1,a)$, Modelo M_1.

	E_0	E_1	E_2	E_3	E_4
A_0	0.9	0.05	0.05	0	0
A_1	0.9	0.05	0.05	0	0
A_2	0.9	0.05	0.05	0	0
A_3	0.9	0.05	0.05	0	0
A_4	0.9	0.05	0.05	0	0
A_5	0.9	0.05	0.05	0	0
A_6	0.95	0.04	0.005	0.0025	0.0025
A_7	0.95	0.04	0.005	0.0025	0.0025
A_8	0.95	0.04	0.005	0.0025	0.0025
A_9	0.95	0.04	0.005	0.0025	0.0025

TABELA 3: $P(s|E_2,a)$, Modelo M_1.

	E_0	E_1	E_2	E_3	E_4
A_0	0.95	0.04	0.01	0	0
A_1	0.95	0.04	0.01	0	0
A_2	0.95	0.04	0.01	0	0
A_3	0.95	0.04	0.01	0	0
A_4	0.95	0.04	0.01	0	0
A_5	0.95	0.04	0.01	0	0
A_6	0.95	0.04	0.005	0.0025	0.0025
A_7	0.95	0.04	0.005	0.0025	0.0025
A_8	0.95	0.04	0.005	0.0025	0.0025
A_9	0.95	0.04	0.005	0.0025	0.0025

TABELA 4: $P(s|E_3,a)$, Modelo M_1.

	E_0	E_1	E_2	E_3	E_4
A_0	0.95	0.04	0.01	0	0
A_1	0.95	0.04	0.01	0	0
A_2	0.95	0.04	0.01	0	0
A_3	0.95	0.04	0.01	0	0
A_4	0.95	0.04	0.01	0	0
A_5	0.95	0.04	0.01	0	0
A_6	0.95	0.04	0.005	0.0025	0.0025

A_7	0.95	0.04	0.005	0.0025	0.0025
A_8	0.95	0.04	0.005	0.0025	0.0025
A_9	0.95	0.04	0.005	0.0025	0.0025

TABELA 5: P(s|E_4,a), Modelo M_1.

	E_0	E_1	E_2	E_3	E_4
A_0	1	0	0	0	0
A_1	1	0	0	0	0
A_2	1	0	0	0	0
A_3	1	0	0	0	0
A_4	1	0	0	0	0
A_5	1	0	0	0	0
A_6	0.95	0.5	0	0	0
A_7	0.95	0.5	0	0	0
A_8	0.95	0.5	0	0	0
A_9	0.95	0.5	0	0	0

TABELA 6: P(s|E_0,a), Modelo M_2.

	E_0	E_1	E_2	E_3	E_4
A_0	0.005	0.01	0.9	0.08	0.005
A_1	0.005	0.01	0.9	0.08	0.005
A_2	0.005	0.01	0.9	0.08	0.005
A_3	0.005	0.01	0.9	0.08	0.005
A_4	0.005	0.01	0.9	0.08	0.005
A_5	0.05	0.01	0.9	0.08	0.005
A_6	0.005	0.01	0.9	0.08	0.005
A_7	0.005	0.01	0.9	0.08	0.005
A_8	0.005	0.01	0.9	0.08	0.005
A_9	0.005	0.01	0.9	0.08	0.005

TABELA 7: P(s|E_1,a), Modelo M_2.

	E_0	E_1	E_2	E_3	E_4
A_0	0.005	0.01	0.9	0.08	0.005
A_1	0.005	0.01	0.9	0.08	0.005
A_2	0.005	0.01	0.9	0.08	0.005
A_3	0.005	0.01	0.9	0.08	0.005
A_4	0.005	0.01	0.9	0.08	0.005
A_5	0.005	0.01	0.9	0.08	0.005
A_6	0.005	0.01	0.9	0.08	0.005
A_7	0.005	0.01	0.9	0.08	0.005
A_8	0.05	0.01	0.9	0.08	0.005
A_9	0.05	0.01	0.9	0.08	0.005

TABELA 8: P(s|E_2,a), Modelo M_2.

	E_0	E_1	E_2	E_3	E_4
A_0	0.005	0.01	0.9	0.08	0.005
A_1	0.005	0.01	0.9	0.08	0.005
A_2	0.005	0.01	0.9	0.08	0.005
A_3	0.005	0.01	0.9	0.08	0.005
A_4	0.005	0.01	0.9	0.08	0.005
A_5	0.005	0.01	0.9	0.08	0.005
A_6	0.005	0.01	0.9	0.08	0.005
A_7	0.005	0.01	0.9	0.08	0.005
A_8	0.005	0.01	0.9	0.08	0.005
A_9	0.005	0.01	0.9	0.08	0.005

TABELA 9: P(s|E_3,a), Modelo M_2.

	E_0	E_1	E_2	E_3	E_4
A_0	0.005	0.01	0.9	0.08	0.005
A_1	0.005	0.01	0.9	0.08	0.005
A_2	0.005	0.01	0.9	0.08	0.005
A_3	0.005	0.01	0.9	0.08	0.005
A_4	0.005	0.01	0.9	0.08	0.005
A_5	0.005	0.01	0.9	0.08	0.005
A_6	0.005	0.01	0.9	0.08	0.005
A_7	0.005	0.01	0.9	0.08	0.005
A_8	0.005	0.01	0.9	0.08	0.005
A_9	0.005	0.01	0.9	0.08	0.005

TABELA 10: P(s|E_4,a), Modelo M_2.

	E_0	E_1	E_2	E_3	E_4
A_0	0.005	0.01	0.9	0.08	0.005
A_1	0.005	0.01	0.9	0.08	0.005
A_2	0.005	0.01	0.9	0.08	0.005
A_3	0.005	0.01	0.9	0.08	0.005
A_4	0.005	0.01	0.9	0.08	0.005
A_5	0.005	0.01	0.9	0.08	0.005
A_6	0.005	0.01	0.9	0.08	0.005
A_7	0.005	0.01	0.9	0.08	0.005
A_8	0.005	0.01	0.9	0.08	0.005
A_9	0.005	0.01	0.9	0.08	0.005

TABELA 11: P(s|E_0,a), Modelo M_3.

	E_0	E_1	E_2	E_3	E_4
A_0	0	0	0	0	1
A_1	0	0	0	0	1
A_2	0	0	0	0	1
A_3	0	0	0	0	1
A_4	0	0	0	0	1
A_5	0	0	0	0	1
A_6	0.0025	0.0025	0.005	0.04	0.95
A_7	0.0025	0.0025	0.005	0.0	0.95
A_8	0.0025	0.0025	0.005	0.04	0.95
A_9	0.0025	0.0025	0.005	0.04	0.95

TABELA 12: P(s|E_1,a), Modelo M_3.

	E_0	E_1	E_2	E_3	E_4
A_0	0	0	0	0	1
A_1	0	0	0	0	1
A_2	0	0	0	0	1
A_3	0	0	0	0	1
A_4	0	0	0	0	1
A_5	0	0	0	0	1
A_6	0.0025	0.0025	0.005	0.04	0.95
A_7	0.0025	0.0025	0.005	0.04	0.95
A_8	0.0025	0.0025	0.005	0.04	0.95
A_9	0.0025	0.0025	0.005	0.04	0.95

TABELA 13: P(s|E_2,a), Modelo M_3.

	E_0	E_1	E_2	E_3	E_4
A_0	0.005	0	0.005	0	0.95
A_1	0.005	0	0.005	0	0.95
A_2	0.005	0	0.005	0	0.95
A_3	0.005	0	0.005	0	0.95
A_4	0.005	0	0.005	0	0.95
A_5	0.005	0	0.005	0	0.95
A_6	0.0025	0.0025	0.005	0.04	0.95
A_7	0.0025	0.0025	0.005	0.04	0.95
A_8	0.0025	0.0025	0.005	0.04	0.95
A_9	0.0025	0.0025	0.005	0.04	0.95

TABELA 14: P(s|E_3,a), Modelo M_3.

	E_0	E_1	E_2	E_3	E_4
A_0	0.0025	0.0025	0.005	0.04	0.95
A_1	0.0025	0.0025	0.005	0.0	0.95
A_2	0.0025	0.0025	0.005	0.04	0.95
A_3	0.0025	0.0025	0.005	0.04	0.95
A_4	0.0025	0.0025	0.005	0.04	0.95
A_5	0.0025	0.0025	0.005	0.04	0.95
A_6	0.0025	0.0025	0.005	0.04	0.95
A_7	0.0025	0.0025	0.005	0.04	0.95
A_8	0.0025	0.0025	0.005	0.04	0.95
A_9	0.0025	0.0025	0.005	0.04	0.95

TABELA15: P(s|E_4,a), Modelo M_3.

	E_0	E_1	E_2	E_3	E_4
A_0	0.0025	0.0025	0.005	0.04	0.95
A_1	0.0025	0.0025	0.005	0.04	0.95
A_2	0.0025	0.0025	0.005	0.04	0.95
A_3	0.0025	0.0025	0.005	0.04	0.95
A_4	0.0025	0.0025	0.005	0.04	0.95
A_5	0.0025	0.0025	0.005	0.04	0.95
A_6	0.0025	0.0025	0.005	0.04	0.95
A_7	0.0025	0.0025	0.005	0.04	0.95
A_8	0.0025	0.0025	0.005	0.04	0.95
A_9	0.0025	0.0025	0.005	0.04	0.95

Finalmente, foi criada uma política pedagógica P (diagramas de transição mostrados nas Figuras 5 a 9).

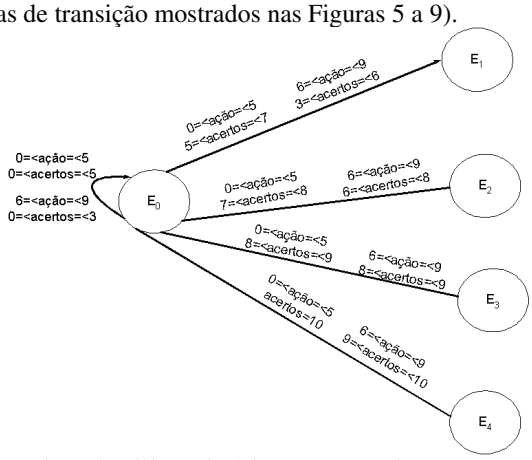


Figura 5. Política Pedagógica P para o Estado E_0.

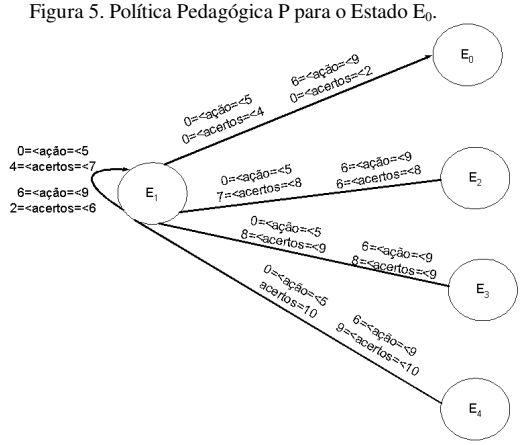


Figura 6. Política Pedagógica P para o Estado E_1.

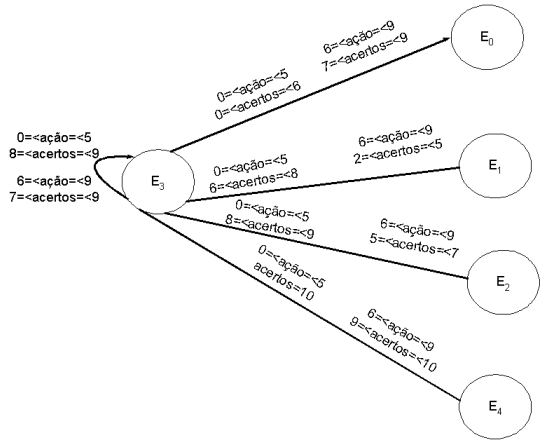


Figura 8. Política Pedagógica P para o Estado E_3.

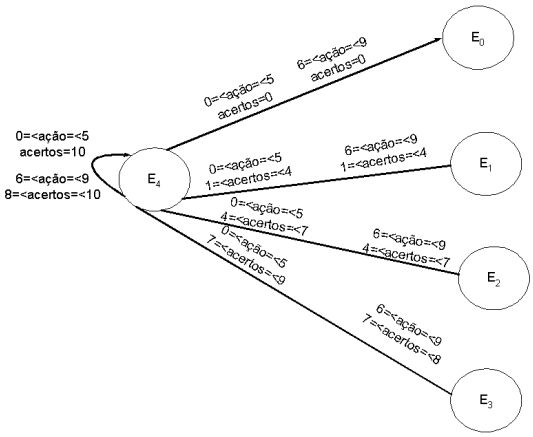


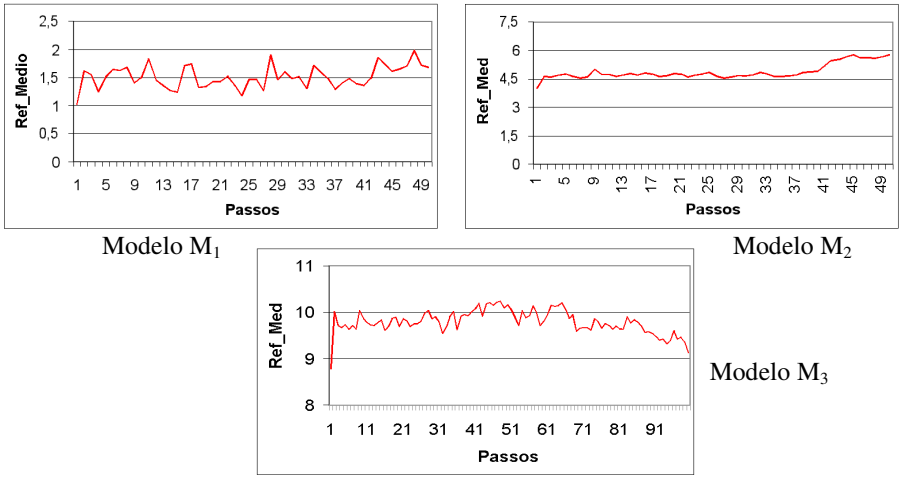
Figura 9. Política Pedagógica P para o Estado E_4.

Os modelos foram submetidos a 1000 passos de treinamento, e de cada conjunto de simulação foi calculada uma média sobre 20 simulações, para obtenção dos dados finais.

4.1. Resultados Obtidos

Os Resultados da Figura 10 mostram as médias dos reforços obtidos durante a simulação da política pedagógica **P**, para cada um dos modelos M_1, M_2 e M_3. Q-Learning consegue convergir para a melhor política de ação de cada modelo, usando α=0.9 e γ=0.9.

A Figura 11 mostra a evolução das médias dos valores de utilidade Q(s,a).Com base nesses valores, o algoritmo consegue definir (através da maximização sobre Q(s,a)) a ação de maior valor de utilidade para cada estado s. Ao longo do tempo ocorreu convergência para uma política ótima de ações (μ) em cada modelo simulado.



Modelo M_1 Modelo M_2

Modelo M_3

Figura 10. Reforços Médios – Modelos M_1, M_2 e M_3.

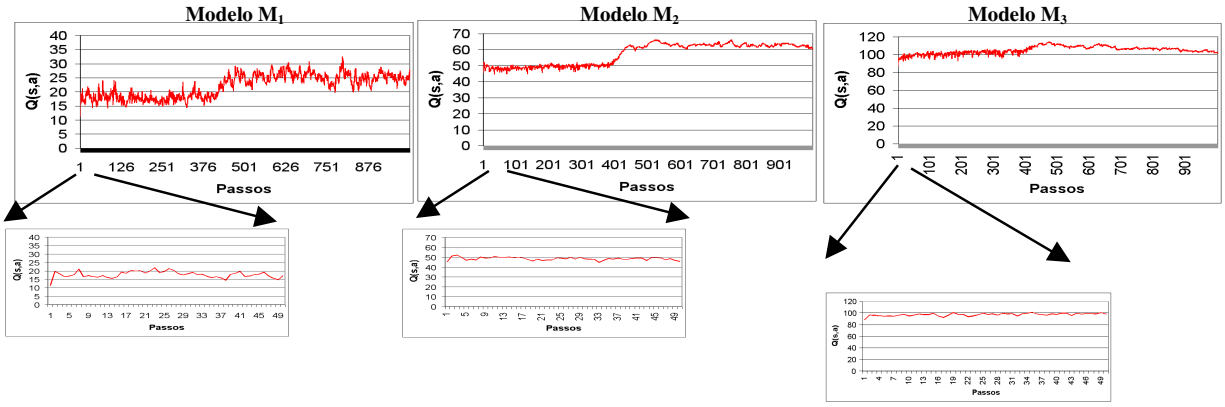
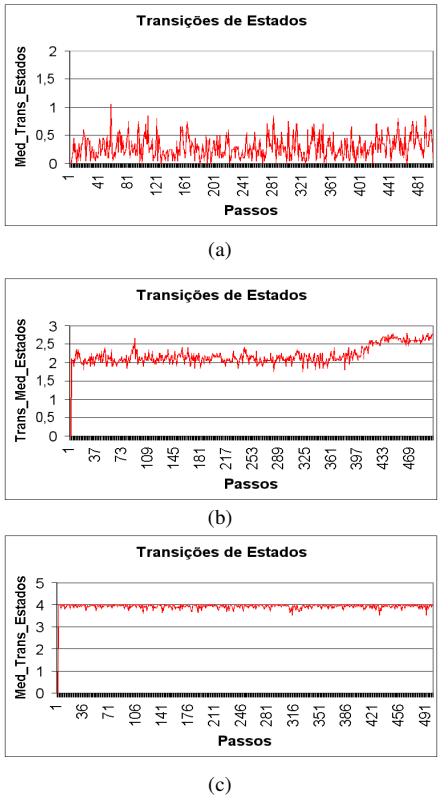


Figura 11. Evolução de Q(s,a) – Modelos M_1, M_2 e M_3.

Os gráficos da Figura 12 mostram as médias das transições entre os estados cognitivos dos modelos simulados. Observe que o STI não conhece os modelos: os resultados são conseqüência da convergência para uma política ótima de ações.



(a)

(b)

(c)

Figura 12. Média das Transições dos Estados: a) M_1, b) M_2 e c) M_3.

5. Conclusão

A utilização do Sistema de Aprendizado por Reforço para Modelagem Autônoma do Aprendiz em um Tutor Inteligente pode oferecer vantagens e possíveis dificuldades que se tornariam desvantagens, dentre as quais:

Vantagens:

- Não requer um modelo do aprendiz.
- Independe do conteúdo apresentado pelo tutor.
- Adapta-se a várias estratégias pedagógicas.
- Possibilidade - através de um log de registro - da continuidade da tutoria do aprendiz a partir do ponto em que esta foi interrompida.

Desvantagens:

- O interfaceamento com tutores já existentes pode ser complicado pelo fato destes não preverem a atualização de parâmetros de algoritmos adaptativos tais como algoritmos de AR.
- Lentidão de convergência do algoritmo – especialmente para grandes espaços de estados e grande quantidade de possíveis ações de tutoria. Esta lentidão pode ser parcialmente compensada pelo uso de variantes baseadas em aproximações compactas [7], estratégias de generalização da experiência [5], ou planos de ação obtidos em simulação [6].

5.2. Proposta para Trabalhos Futuros

Uma contribuição deste trabalho foi o resultado positivo obtido na simulação com o uso do algoritmo Q-Learning aplicado a um agente tutor que inclui na sua arquitetura o módulo diagnóstico, para que desta forma se consiga dar um passo importante na questão da modelagem autônoma do aprendiz em um tutor inteligente. Acredita-se que este trabalho venha contribuir para formalização computacional do problema de modelagem do estado cognitivo do aprendiz de forma dinâmica. Sendo assim, os resultados viabilizam a utilização de tutores inteligentes que utilizam o Sistema de AR em ambientes on-line, como por exemplo Internet.

Outra contribuição deste trabalho é, deixar dados para estudos de viabilidade de desempenho no que se refere a redução do tempo de convergência do algoritmo, para uso mais efetivo de tutores inteligentes com o Sistema de AR .

Além das contribuições citadas acima, podemos evidenciar possibilidades de trabalhos futuros, como:

- O interfaceamento de tutores inteligentes já existentes no mercado com o Sistema de AR.
- Desenvolvimento de um sistema de Tutoria Inteligente, onde se possa utilizar múltiplas estratégias, independente do domínio.
- Com a estrutura desenvolvida neste trabalho, adaptá-la para o uso de multiagentes.

6. Referências

[1] Bellman, R. E. *Dynamic Programming*. Princeton: Princeton University Press, 1957.

[2] Giraffa, L.M.M.: "Uma Arquitetura de Tutor utilizando Estados Mentais", Rio Grande do Sul: PUC-RS,1999. Tese de Doutorado.

[3] Marietto, M. G.:"Definição Dinâmica de Estratégias Instrucionais em Sistema de Tutoria Inteligente: Um Abordagem Multiagentes na WWW", São Jose dos Campos:ITA,2000Tese de Doutorado.

[5] Ribeiro, C.H.C. "Embedding a priori Knowledge in Reinforcement Learning", Journal of Intelligent and Robotic Systems. Dordrecht, Holanda: , v.2, n.1, p.51 - 71, 1998.

[6] Sutton, R.and Barto, A. Reinforcement Learning: An Introduction. MIT Press, 1998.
Ribeiro, C.H.C. "Embedding a priori Knowledge in Reinforcement Learning", Journal of Intelligent and Robotic Systems. Dordrecht, Holanda: , v.2, n.1, p.51 - 71, 1998.

[7] Tsitsiklis, J. e Bertsekas, D. (1996). *Neuro-Dynamic Programming*. Athena Scientific, 1996.

[8] Viccari, R.M.: "Um Tutor Inteligente para a Programação em Lógica – Idealização,Projeto e Desenvolvimento",Portugal Universidade de Coimbra,1990,Tese de Doutorado.

[9] Watkins, C. J. C. H.:" Learning from Delayed Rewards.", PhD thesis, University of Cambridge,1989.

[10] Watkins, C. J.C.H., and Dayan, P. "Q-leaning". Machine Learning 8(3/4):279-292, 1992.