UNIVERSIDADE FEDERAL DOS VALES DO JEQUITINHONHA E MUCURI

Programa de Pós-Graduação Em Educação

Éverton de Oliveira Paiva

MELHORIA NA CONVERGÊNCIA DO ALGORIMO *Q-LEARNING* NA APLICAÇÃO DE SISTEMAS TUTORES INTELIGENTES

Diamantina

2016

Éverton de Oliveira Paiva

MELHORIA NA CONVERGÊNCIA DO ALGORIMO *Q-LEARNING* NA APLICAÇÃO DE SISTEMAS TUTORES INTELIGENTES

Dissertação apresentada ao programa de Pós-Graduação em Educação da Universidade Federal dos Vales do Jequitinhonha e Mucuri, como requisito para obtenção do título de Mestre.

Orientador: Prof. Dr. Marcus Vinicius Carvalho Guelpeli

Diamantina

2016

Ficha Catalográfica – Serviço de Bibliotecas/UFVJM
Bibliotecário Anderson César de Oliveira Silva, CRB6 – 2618.

P149m

Paiva, Éverton de Oliveira

Melhoria na convergência do algoritmo Q-Learning na aplicação de sistemas tutores inteligentes / Éverton de Oliveira Paiva. – Diamantina, 2016.

75 p. : il.

Orientador: Marcus Vinícius Carvalho Guelpeli

Dissertação (Mestrado Profissional – Programa de Pós-Graduação em Educação) - Universidade Federal dos Vales do Jequitinhonha e Mucuri.

1. Sistemas Tutores Inteligentes. 2. Modelagem autônoma. 3. Q-Learning. 4. Melhoria na convergência. 5. Metaheurísticas. I. Título. II. Universidade Federal dos Vales do Jequitinhonha e Mucuri.

CDD 371.334

Elaborada com dados fornecidos pelo(a) autor(a).

Éverton de Oliveira Paiva

MELHORIA NA CONVERGÊNCIA DO ALGORIMO *Q-LEARNING* NA APLICAÇÃO DE SISTEMAS TUTORES INTELIGENTES

Dissertação apresentada ao programa de Pós-Graduação em Educação da Universidade Federal dos Vales do Jequitinhonha e Mucuri, como requisito parcial para obtenção do título de Mestre.

Orientador: Prof. Dr. Marcus Vinicius Carvalho Guelpeli

Data de aprovação 16/08/2016.

Prof. Dr. MARCUS VINÍCIUS CARVALHO GUELPELI
Faculdade de Ciências Exatas - UFVJM

Prof. Dr. ALEXANDRE RAMOS FONSECA
Instituto de Ciência e Tecnologia - UFVJM

Prof. Dr. EULER GUIMARAES HORTA
Instituto de Ciência e Tecnologia - UFVJM

Diamantina

Dedico esse trabalho a minha família,
pelo apoio, pelo amor, pelo exemplo, pela inspiração.

Dedico também ao meu amor, Patrícia,
pela compreensão de tantos momentos tomados pelo trabalho.

AGRADECIMENTOS

Ao meu orientador, professor Marcus, pela dedicação, pela disponibilidade, pela paciência e pela confiança no nosso trabalho.

Aos membros da banca, pelas críticas construtivas sempre visando o enriquecimento do trabalho.

Aos colegas de trabalho pelo incentivo à ingressar no programa.

Aos colegas do programa pelas experiências compartilhadas, pela união e pela presteza.

Aos professores do programa multidisciplinar pela generosidade em acolher os alunos das mais diversas áreas do conhecimento, com as experiências mais distintas e nos fazer caminhar lado a lado do início ao fim dessa jornada.

À UFVJM pela sensibilidade em incentivar ao seu corpo técnico à constante capacitação profissional em busca de prestar um ensino superior de excelência.

It must have been cold there in my shadow
To never have sunlight in your face
And you can content to let me shine
You always walked a step behind.

I was the one with all the glory
While you were the one with all the strength
Only a face without a name
I never once heard you complain.

Did you ever know that you're my hero
And everything I would like to be
I can fly higher than an eagle
Cause you are the wind beneath my wings.

It might have appeared to go unnoticed
but I've got it all here in my heart
I want you to know, I know the truth, yes I do
I would be nothing without you.

Did you ever know that you're my hero
And everything I would like to be
I can fly higher than an eagle
Cause you are the wind beneath my wings.

Did you ever know that you're my hero
And everything I would like to be
I can fly higher than an eagle
Cause you are the wind beneath my wings.

Did you ever know that you're my hero
And everything I would like to be
I can fly higher than an eagle
Cause you are the wind beneath my wings.

Cause you are the wind beneath my wings.
Cause you are the wind beneath my wings.
Cause you are the wind beneath my wings.

(The Wind Beneath My Wings. Sonata Arctica)

RESUMO

O uso sistemas computacionais como complemento ou substituição da sala de aula é cada vez mais comum na educação e os Sistemas Tutores Inteligentes (STIs) são uma dessas alternativas. Portanto é fundamental desenvolver STIs capazes tanto de ensinar quanto aprender informações relevantes sobre o aluno através de técnicas de inteligência artificial. Esse aprendizado acontece por meio da interação direta entre o STI e o aluno que é geralmente demorada. Esta dissertação apresenta a inserção da metaheurísticas Lista Tabu e GRASP com o objetivo de acelerar esse aprendizado. Para avaliar o desempenho dessa modificação, foi desenvolvido um simulador de STI. Nesse sistema, foram realizadas simulações computacionais para comparar o desempenho da tradicional política de exploração aleatória e as metaheurísticas propostas Lista Tabu e GRASP. Os resultados obtidos através dessas simulações e os testes estatísticos aplicados indicam fortemente que a introdução de meta-heurísticas adequadas melhoram o desempenho do algoritmo de aprendizado em STIs.

Palavras chave: Sistemas Tutores Inteligentes. Modelagem Autônoma. *Q-Learning*. Melhoria na convergência. Metaheurísticas. Lista Tabu. GRASP.

ASTRACT

Using computer systems as a complement or replacement for the classroom experience is an increasingly common practice in education and Intelligent Tutoring Systems (ITS) are one of these alternatives. Therefore, it is crucial to develop ITS that are capable of both teaching and learning relevant information about the student through artificial intelligence techniques. This learning process occurs by means of direct, and generally slow, interaction between the ITS and the student. This dissertation presents the insertion of meta-heuristic Tabu search and GRASP with the purpose of accelera ting learning. An ITS simulator was developed to evaluate the performance of this change. Computer simulations were conducted in order to compare the performance of traditional randomized search methods with the meta-heuristic Tabu search. Results obtained from these simulations and statistical tests strongly indicate that the introduction of meta-heuristics in exploration policy improves the performance of the learning algorithm in ITS.

Keywords: Intelligent tutoring system. Autonomous model. Q-Learning. Convergence improvement. Tabu search. GRASP.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AAC	Aquisição Automatizada de Conhecimento
AR	Aprendizagem por Reforço
AVA	Ambiente Virtual de Aprendizagem
GRASP	*Greedy Randomized Adaptive Search Procedure*
IA	Inteligência Artificial
IAC	Instrução Assistida por Computador
IACI	Instrução Assistida por Computador Inteligente
IDE	*Integrated Development Environment*
LDA	*Level Determinant Agent*
LRC	Lista Restrita de Candidatos
STI	Sistema Tutor Inteligente

SUMÁRIO

1 INTRODUÇÃO

No final da década de 50 e início da década de 60, pesquisadores tais como Alan Turing, Marvin Minsky, John McMarthy e Allen Newell acreditavam que os computadores poderiam "pensar", tal qual os seres humanos (NORVING; RUSSELL, 2013). No entanto, esse fato não aconteceu. Acreditava-se que o principal obstáculo à realização desse objetivo era a necessidade de criação de computadores maiores e mais rápidos. A criação de máquinas capazes de pensar parecia suficiente para acreditar-se que elas seriam capazes de realizar tarefas associadas ao pensamento humano como, por exemplo, a instrução. A máquina, sem a interferência humana, seria capaz de ensinar.

O início da utilização de sistemas computacionais se dá nesse período. Chamados de sistemas de Instrução Assistida por Computador (IAC), os primeiros sistemas tinham poucas capacidades cognitivas e obrigavam o aluno a atuar de forma passiva, isto é, a sua participação resumia-se à seleção de alternativas (VICCARI, 1989). O modelo da apresentação do conteúdo não podia ser alterado: o sistema agia da mesma forma com todos os alunos (ensino programado linearmente).

A partir da década de 60, a técnicas de Inteligência Artificial (IA) foram adicionadas a esses sistemas. A evolução desses sistemas deu origem aos sistemas de Instrução Assistida por Computador Inteligente (IACI), em que a IA desempenha um papel relevante por permitir não só uma maior flexibilidade, mas também possibilitar a participação ativa do aluno e do sistema. Essa evolução gerou um ambiente cooperante para o ensino e a aprendizagem (VICCARI, 1989). O termo Sistema Tutor Inteligente (STI) foi adotado para diferenciar esses novos sistemas de seus antecessores, sistemas de IAC.

Atualmente o termo STI assume um papel mais amplo, abrangendo qualquer programa que possui alguma inteligência e pode ser utilizado em aprendizagem (FREEDMAN, 2000). Os STIs se diferenciam dos sistemas de IAC pela adição de estratégias de ensino do conhecimento e por manterem um modelo atualizado das atividades do aprendiz. Agregar técnicas de Inteligência Artificial aos tradicionais sistemas IAC significa trabalhar de forma interdisciplinar com as conquistas que outras áreas de pesquisa obtiveram em relação ao conhecimento da comunicação inteligente, tais como os avanços da psicologia e da pedagogia (GIRAFFA, 1999).

Os STIs pertencem a categoria de software educacionais que se baseiam na aprendizagem interativa. Nesse contexto, o aluno passa a ser o centro do processo ensino-aprendizagem, deixando de ser passivo e tornando-se um ser ativo no processo, além de tornar

relevante o seu conhecimento atual e as suas características de aprendizado (JESUS, 2009). Por esse motivo, existe uma preocupação em gerar STIs capazes de interagir com o aluno, afim de gerar o modelo cognitivo desse aluno. Dessa forma, através do modelo gerado será possível selecionar e aplicar a técnica pedagógica mais adequada.

A evolução das tecnologias de comunicação bem como dos sistemas computacionais ampliou as possibilidades de utilização dos sistemas em aprendizagem, inclusive em modelos de ensino não-presencial. Os sistemas de *E-Learning* (do inglês *electronic learning*, "aprendizagem eletrônica") são oriundos dessa evolução. No entanto, os tradicionais sistemas de *E-Learning* (LI; ZHOU, 2015) foram criticados devido suas limitações. Esse sistemas sempre apresentam os mesmos materiais e tópicos para os alunos, independentemente de seu conhecimento prévio, habilidades de aprendizagem e níveis de entendimento sobre o assunto. Em contrapartida, os STIs empregam uma base que contém conhecimento especialista no assunto, estratégias de ensino e heurísticas. Esses sistemas ser capazes de selecionar materiais de ensino relevantes de forma dinâmica e, portanto, escolher diferentes caminhos pedagógicos, exemplos e exercícios para diferentes alunos.

Esses sistemas oferecem flexibilidade na apresentação do material e maior habilidade para responder às necessidades do aluno. Procuram, além de ensinar, aprender informações relevantes sobre o aluno, proporcionando um aprendizado individualizado. STIs têm sido apresentados como altamente eficientes para a melhora do desempenho e motivação dos alunos (PALOMINO, 2013). Para que eles possuam essa capacidade, são utilizadas técnicas de IA como Aprendizagem por Reforço (AR).

A AR é um formalismo da IA que permite a um agente computacional aprender a partir da interação com o ambiente no qual se encontra inserido (SUTTON; BARTO, 1998). Por sua vez, esta aprendizagem de uma Política ótima (ou quase ótima) para um determinado ambiente decorre do uso das recompensas observadas (NORVING; RUSSELL, 2013). Essa técnica é atraente para solucionar uma variedade de problemas quando não existem modelos disponíveis, a priori, já que seus algoritmos têm a convergência para uma situação de equilíbrio garantida (LITTMAN; SZEPESVARI, 1996), além de permitirem o aprendizado de estratégias de controle adequadas. Dentre esses algoritmos de AR encontra-se o algoritmo *Q-Learning* (WATKINS, 1986), utilizado no presente trabalho.

1.1 Problema

Em algoritmos de aprendizagem por reforço o aprendizado acontece por meio de interação direta entre o agente e o ambiente. A convergência dos algoritmos de AR só pode ser atingida após uma extensiva exploração do espaço de estados-ações, que é geralmente demorada.

1.2 Motivação

Não se encontrou na bibliografia pesquisada trabalhos com o objetivo de solucionar o problema da lenta convergência de algoritmos de aprendizagem por reforço na área de sistemas tutores inteligentes.

Sistemas Tutores Inteligentes com característica de modelagem autônoma podem ser utilizados como ferramentas auxiliares de aprendizagem em diversas áreas, como por exemplo na área de Educação a Distância. Nem sempre existem modelos de aprendiz disponíveis, portanto a convergência rápida do algoritmo pode viabilizar a utilização desses sistemas.

1.3 Hipótese

A velocidade de convergência de um algoritmo de AR pode ser acelerada através de algumas técnicas. Através da inserção de metaheurísticas seria possível melhorar o tempo de convergência do algoritmo Q-Learning na aplicação de STI's com característica de modelagem autônoma do aprendiz?

1.4 Contribuições

As contribuições da introdução de metaheurísticas, no modelo proposto neste trabalho, em sistemas tutores inteligentes com a característica de modelagem autônoma do aprendiz apresentam a seguinte finalidade:

- Implementar técnicas de aceleração do tempo de convergência do algoritmo *Q-Learning* em STIs, analisar os resultados e realizar testes estatísticos dos resultados obtidos em relação à implementação original.
- Propor modificações no algoritmo apresentado por Guelpeli, Omar e Ribeiro (2004), a fim de se obter redução no tempo de convergência no algoritmo *Q-Learning* na aplicação de sistemas tutores inteligentes.

- Viabilizar a utilização desse tipo de sistema tutor inteligente em Ambientes Virtuais de Aprendizagem (AVA) para sistemas de Ensino à Distância.
- Pretende-se avaliar o desempenho dessas modificações propostas em relação ao algoritmo original.
- Espera-se que os resultados obtidos pelas modificações propostas indiquem direções de melhoria do algoritmo originalmente proposto. Através dessas indicações, propor novas modificações utilizando os resultados dos primeiros experimentos como base.
- A convergência mais rápida pode tornar viável a utilização de STIs em ambientes onde não existem modelos disponíveis.

1.5 Metodologia de Pesquisa

Trata-se de um trabalho que envolve elaboração e testes de algoritmos. Tais atividades demandam uso extensivo de recursos computacionais para implementação e execução das simulações, análise estatística e elaboração de resultados comparativos. Foi utilizado suporte bibliográfico físico e virtual.

Inicialmente, realizou-se adaptação do programa simulador de Sistema Tutor Inteligente (GUELPELI; OMAR; RIBEIRO, 2004) desenvolvido originalmente sob programação estruturada e utilização em ambiente de terminal para linguagem C++. O novo programa foi desenvolvido utilizando conceitos de programação Orientada à Objetos, através da IDE Qt Creator. Essa IDE permite a utilização multiplataforma do aplicativo, além de fornecer componentes de interface para visualização dos parâmetros a serem analisados pelo simulador como rótulos, barras de progresso, tabelas e gráficos, todos atualizados em tempo de execução do programa.

Após esta etapa, definiu-se as metaheurísticas que foram utilizadas para buscar a melhoria do tempo de convergência do algoritmo e aumento da métrica de qualidade das ações escolhidas pelo simulador de sistema tutor inteligente. Foram adotadas as metaheurísticas Lista Tabu e GRASP. Após essa escolha, o próximo passo realizado foi a implementação dessas metaheurísticas na fase de exploração do algoritmo, detalhada no capítulo 3.

Com a implementação das metaheurísticas realizada, foi possível executar simulações e identificar a parametrização que apresentou os melhores resultados nas simulações de cada metaheurística. Após essa etapa, foram realizadas as simulações comparativas entre o algoritmo original e as metaheurísticas selecionadas, desta forma, produzindo resultados comparativos.

1.6 Estrutura do Texto

No Capítulo 2, Fundamentação Teórica, serão apresentados conceitos de sistemas tutores inteligentes, aprendizagem por reforço e metaheurísticas. Dentre as metaheurísticas, serão detalhadas a Busca Tabu e GRASP. Ao final do capítulo são apresentados trabalhos correlatos.

No Capítulo 3, Aceleração de Convergência Através de Metaheurísticas, o modelo proposto neste trabalho é apresentado. O capítulo ainda traz o modelo de funcionamento da exploração através das metaheurísticas Busca Tabu e GRASP bem como suas justificativas de parametrização.

No Capítulo 4, Simulador, são exibidas as telas do programa, as configurações disponíveis para as simulações, o acompanhamento dos resultados parciais das simulações e os arquivos de saída com os resultados.

O Capítulo 5, Resultados, apresentará e discutirá os resultados comparativos entre o algoritmo original e as metaheurísticas adotadas para cada um dos 3 modelos de aprendiz adotados nas simulações. Ainda encontram-se nesse capítulo a comprovação da hipótese e a análise dos resultados estatísticos.

O Capítulo 6, Conclusões, apresentará as contribuições deste trabalho, suas limitações e, finalmente, sugestão de trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão apresentados os conceitos de Sistemas Tutores Inteligentes, Aprendizagem por Reforço e Meta-Heurísticas. Na seção dedicada aos conceitos de Meta-Heurísticas, são detalhadas as meta-heurísticas Busca Tabu e GRASP que são utilizadas na metodologia deste trabalho.

2.1 Sistemas Tutores Inteligentes

Na década de 1950 os computadores começaram a ser utilizados em aprendizagem. Inicialmente através dos programas IAC (Instrução Assistida por Computador), o conteúdo era transmitido de maneira sequencial, em uma estrutura previamente determinada e sem a capacidade de adaptação individual. A Inteligência Artificial foi adicionada aos sistemas IAC, originando os sistemas IACI (Instrução Assistida por Computador Inteligente) agora com a capacidade de representação do conhecimento e classificação cognitiva, aumentando o grau de "inteligência" dos sistemas educacionais.

Os Sistemas Tutores Inteligentes (STI's) são uma evolução dos sistemas IAC e se diferenciam de seus antecessores pela adição de estratégias de ensino do conhecimento e por manterem um modelo atualizado das atividades do aprendiz. Segundo Giraffa (1999) os STI's são uma modalidade dos IACI e modela o aprendiz de forma individualizada. Quando esse modelo é "fraco", ou seja, o perfil do aprendiz não está bem modelado, os estados cognitivos não têm uma representação fidedigna, os STI's são denominados como assistentes. Quando é "forte", ou seja, uma representação exata dos seus estados mentais, são denominados então como tutores. Os STI's possibilitam ao aprendiz a capacidade de aprender com o tutor, servindo este como guia no processo; ele deve se adaptar ao aprendiz, e não o contrário, como acontece no método tradicional. Com isso, é necessário um modelamento do aprendiz, para que os STI's possam saber o que ensinar, a quem ensinar e como ensinar. Ele deve ser capaz de mudar o nível de entendimento para responder às entradas do aprendiz, em vários níveis, podendo mudar as estratégias pedagógicas, adaptando-se de forma individualizada, de acordo como ritmo e as características de cada aprendiz.

Em uma definição mais ampla, Freedman (2000) define um Sistema Tutor Inteligente como qualquer programa que possui alguma inteligência e pode ser utilizado em aprendizagem.

Um tutor inteligente necessita explorar os conteúdos, possuir vários planos de ensino e um modelo para guiar a apresentação do conteúdo, ser sensível às necessidades do

utilizador adequando-se às necessidades individuais, dominar o máximo possível o assunto que ensina, possuir conhecimento para tentar resolver situações não previstas nas regras existentes e aprender com tais situações, possuir características de ensino assistido. Possuir mecanismos para a depuração inteligente e a orientação na detecção e eliminação de falhas, permitir a simulação automática e conduzida de problemas, além de possuir memória retroativa que descreva o raciocínio utilizado pelo aluno e pelo tutor durante a exploração de determinado conteúdo (VICCARI, 1989).

Os Sistemas Tutores Inteligentes, em sua arquitetura clássica (VICCARI, 1989), apresentam os seguintes módulos: Módulo do Aprendiz, Módulo de Tutoria, Módulo de Domínio e Módulo de Interface. Essa arquitetura está representada na Figura 1:

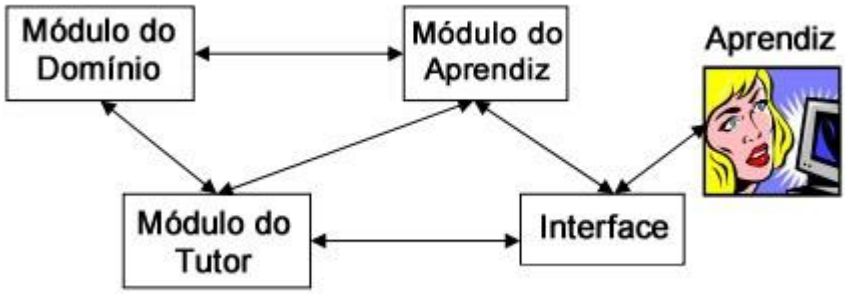

Figura 1 - Arquitetura clássica de um STI (GAVÍDIA; ANDRADE, 2003).

O **Módulo do Aprendiz** é responsável por captar o atual estado cognitivo do aprendiz, individualizando o ensino, representando aspectos do comportamento e conhecimento do aprendiz. Deve ser capaz de detectar erros cometidos pelos aprendizes e verificar mudanças no perfil do aprendiz, gerando processo de diagnóstico.

O **Módulo do Tutor** centraliza as ações dos outros módulos. É neste que se encontram as estratégias didáticas e pedagógicas adotadas por cada STI. Essas estratégias são responsáveis pelo planejamento instrucional, que objetiva apresentar um conteúdo ao aprendiz de acordo com um diagnóstico de suas deficiências e eficiência de conhecimento em um dado domínio.

O **Módulo do Domínio** contém o conhecimento do especialista, que será mostrado ao aprendiz e atuará como base do conhecimento para questões, respostas e tarefas; e também com o padrão para avaliação do desempenho do aprendiz.

A **Interface** é o mecanismo pelo qual o aprendiz se comunica com o tutor. O objetivo geral da interface do usuário é permitir interações, através de teclado, mouse, monitor, ocorram enviando e recebendo informações do aluno concretizando a comunicação de maneira eficaz.

2.2 Aprendizagem por Reforço

Turing (1950) propõe que é mais vantajoso construir máquinas com capacidade de aprendizagem e depois ensiná-las. Dessa maneira seria possível operar em ambientes inicialmente desconhecidos e, com o passar do tempo, se tornar mais competente do que seu conhecimento inicial.

Um agente é tudo que pode ser considerado capaz de perceber seu ambiente por meio de sensores e de agir sobre esse ambiente por intermédio de atuadores (NORVING; RUSSELL, 2013). Essa ideia é ilustrada na Figura 2:

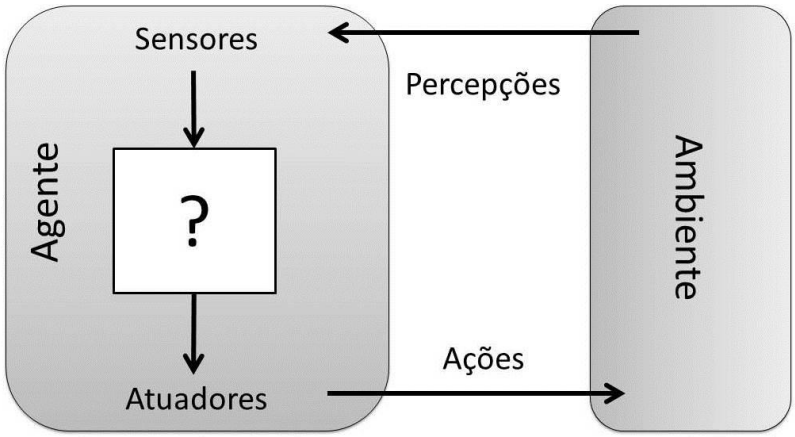

Figura 2 - Agentes interagem com ambientes por meio de sensores e atuadores (NORVING; RUSSELL, 2013).

Um agente de aprendizado pode ser dividido em quatro componentes conceituais (NORVING; RUSSELL, 2013), como pode ser visualizado na Figura 3. A distinção mais importante se dá entre o elemento de aprendizado, responsável pela execução de aperfeiçoamentos, e o elemento de desempenho, responsável pela seleção de ações externas. O elemento de desempenho recebe percepções e decide sobre ações. O elemento de aprendizado utiliza realimentação do crítico sobre como o agente está funcionando e determina de que maneira o elemento de desempenho deve ser modificado para funcionar melhor no futuro.

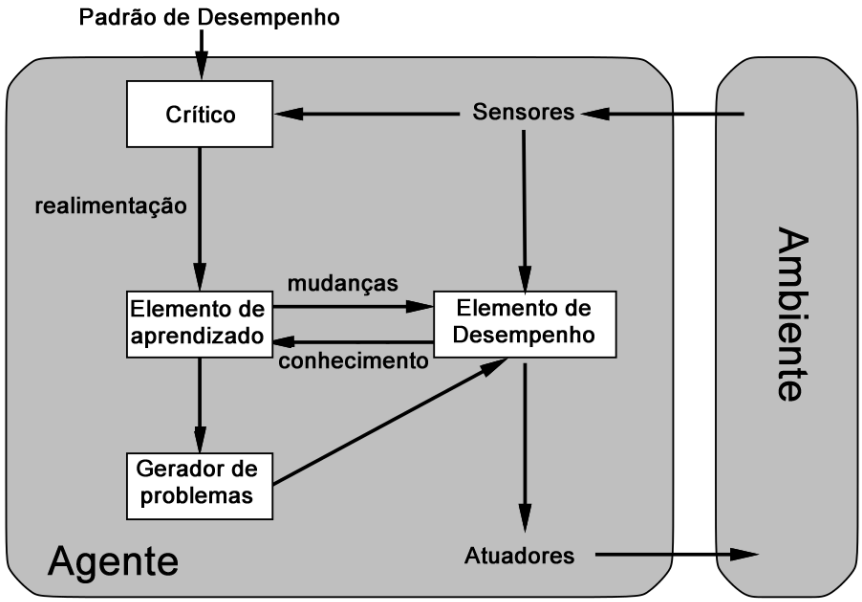

Figura 3 - Um modelo geral de agentes com aprendizagem (NORVING; RUSSELL, 2013).

No aprendizado por reforço (AR), concebe-se algoritmos computacionais para transformar situações do mundo em ações, de forma a maximizar a medida de recompensa (LUGER, 2013). O agente não sabe diretamente o que fazer ou qual ação tomar; em vez disso, ele descobre por meio de exploração quais ações oferecem maior recompensa. As ações do agente afetam não apenas a recompensa imediata, mas também têm impacto sobre as ações e eventuais recompensas subsequentes.

O aprendizado por reforço deve ainda, considerar o compromisso entre usar apenas o que ele sabe no momento ou continuar explorando o mundo. Para otimizar as suas possibilidades de recompensa, o agente não deve apenas fazer o que ele já sabe, mas também explorar aquelas partes do seu mundo que ele ainda desconhece. O agente deve explorar uma série de opções e, ao mesmo tempo, favorecer aquelas que pareçam ser as melhores.

A aprendizagem por reforço (SUTTON; BARTO, 1998) é um formalismo da inteligência artificial que permite a um agente computacional aprender a partir da interação com o ambiente no qual se encontra inserido. Por sua vez, esta aprendizagem de uma política ótima (ou quase ótima) para um determinado ambiente decorre do uso das recompensas observadas (NORVING; RUSSEL, 2013). É uma técnica atraente para solucionar uma variedade de problemas quando não existem modelos disponíveis, a priori, uma vez que seus algoritmos têm a convergência para uma situação de equilíbrio garantida (LITTMAN; SZEPESVARI, 1996), além de permitirem o aprendizado de estratégias de controle adequadas. Neste caso, o aprendizado acontece por meio de interação direta entre o agente e o ambiente.

Tido como o mais popular algoritmo de AR, o algoritmo *Q-Learning* foi proposto como uma maneira de aprender iterativamente a política ótima ($\pi *$) quando o modelo do sistema não é conhecido (WATKINS, 1989). No *Q-Learning* a escolha de uma ação é baseada em uma função de utilidade que mapeia estados e ações a um valor numérico. O *Q-Learning* produz uma atualização dos valores Q na direção $max_a Q^u_{t\,(s_{t+1},a)}$, apresentado em Algoritmo 1:

1	Inicialize $Q(s,a)$
2	Para cada instante t repita:
3	Observe estado s_t e escolha ação $a_t$de acordo com a política de ações (μ);
4	Observe o estado s_{t+1} e atualize $Q^u_{t\,(s_t,a_t)}$de acordo com: $Q^u_{t+1(s_t,a_t)=}\, Q^u_{t\,(s_t,a_t)} + \alpha\left[r(s_t) + \gamma max_a Q^u_{t\,(s_{t+1},a)} - Q^u_{t\,(s_t,a_t)}\right];$
5	Até t igual ao limite de passos.

Algoritmo 1 – Algoritmo *Q-Learning* (WATKINS, 1989).

Onde pode-se definir:

- $Q^u_{t+1(s_t,a_t)}$ é o valor (qualidade) da ação a_t no estado s_t, seguindo a política de ações (μ).
- $r(s_t)$ é o esforço imediato recebido no estado s_t.
- α é a taxa de aprendizagem.
- γ é a taxa de desconto.
- t é a sequência discreta de passos no tempo, ou seja, $t = 0, 1, 2, 3, ...$
  $max_a Q^u_{t\,(s_{t+1},a)}$ política de maximização, que escolhe a ação com maior valor de utilidade no estado futuro.
- O fator γ (entre 0 *e* 1), quanto mais perto de 1, mais importância é dada aos reforços mais distantes no tempo.

A convergência dos algoritmos de Aprendizagem por reforço só pode ser atingida após uma extensiva exploração do espaço de estados-ações, que é geralmente demorada. Entretanto, a velocidade de convergência de um algoritmo de aprendizagem por reforço pode ser acelerada através de algumas técnicas. Este trabalho, apresenta a inserção das metaheurística Busca Tabu e GRASP como estratégias de exploração do algoritmo *Q-Learning* com o objetivo de alcançar essa aceleração. A seção 2.3 Metaheurísticas abordará de maneira aprofundada esse assunto.

2.2.1 Sistemas Tutores Inteligentes com Modelagem Autônoma

Para Giraffa (1999), as dificuldades em implementar o que se conhece a respeito do processamento de informação humano têm apresentado restrições que implicam na perda do modelo mental humano já conhecido em favor de um modelo computacional, gerando perdas significativas de qualidade. O que é considerado como teoria na área de Educação, nem sempre satisfaz os requisitos de uma teoria formal na área de Ciência da Computação, daí a dificuldade encontrada em definir o modelo do aprendiz em um STI.

Dadas essas dificuldades, Guelpeli, Omar e Ribeiro (2004) propõem alterações na arquitetura clássica dos STIs, adicionando um novo módulo de diagnóstico. Neste módulo são aplicadas técnicas de AR, através do algoritmo *Q-Learning* (WATKINS, 1989), o que possibilita modelar autonomamente o aprendiz. Um valor de utilidade é calculado baseado em uma tabela de pares estado-ação, a partir da qual o algoritmo estima reforços futuros que representam os estados cognitivos do aprendiz. A melhor política a ser usada pelo tutor para qualquer estado cognitivo do aprendiz é disponibilizada pelo próprio algoritmo de AR, sem que seja necessário um modelo explícito do aprendiz. Essa nova proposta de arquitetura está representada na Figura 4:

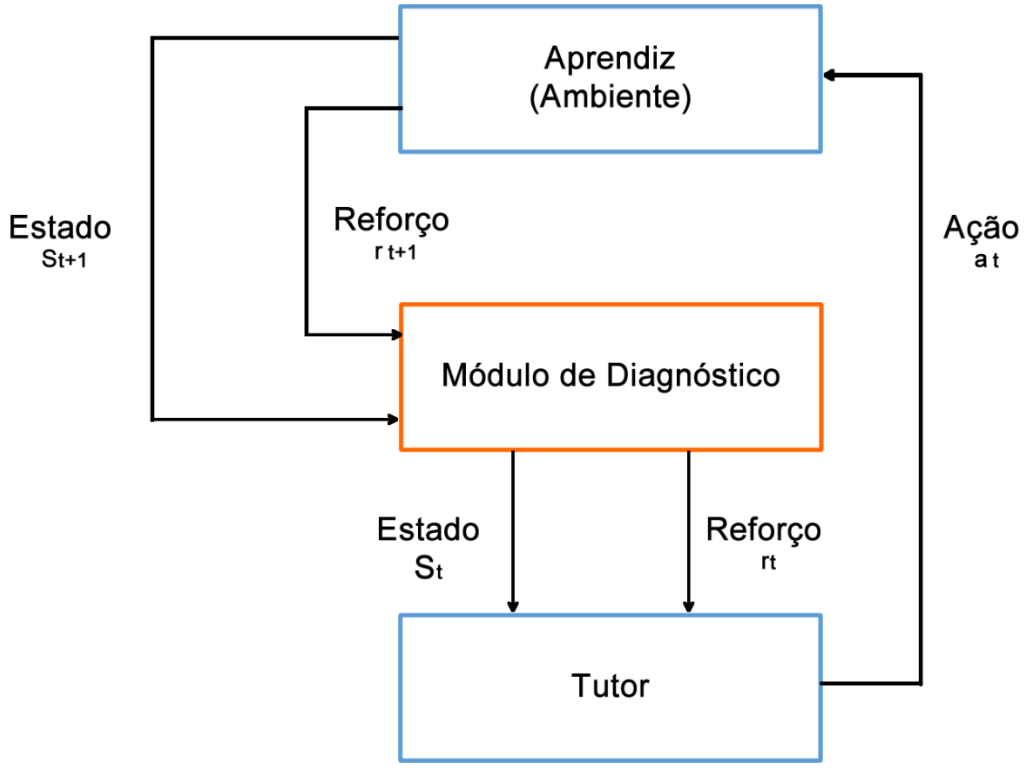

Figura 4 - Identificação dos módulos do Sistema de Aprendizado por reforço para Modelagem Autônoma do Aprendiz em Tutor Inteligente (GUELPELI; OMAR; RIBEIRO, 2004).

O sistema realiza uma classificação inicial do estado cognitivo do aprendiz. A partir desse ponto, para cada iteração, o sistema escolhe uma determinada ação, aplica testes, avalia os resultados, reclassifica o estado cognitivo do aprendiz e atualiza a tabela de acordo com o desempenho de cada ação escolhida. Com o passar das iterações o sistema identifica quais as ações obtiveram maior desempenho para cada estado cognitivo do aprendiz. Para melhor entendimento do sistema a Figura 5 representa esse funcionamento:

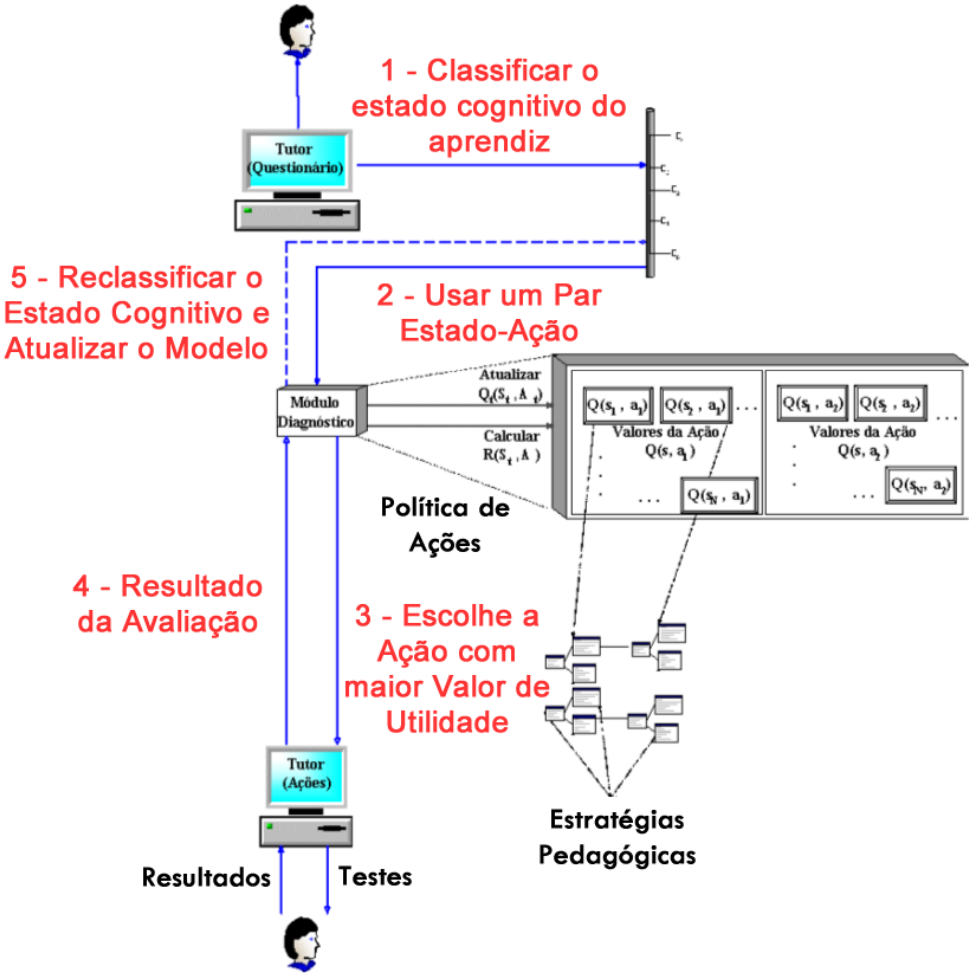

Figura 5 - Sistema com a técnica de Aprendizagem por Reforço (GUELPELI; OMAR; RIBEIRO, 2004).

O passo a passo do funcionamento desse sistema é representado no Algoritmo 2:

1	O sistema classifica o estado cognitivo do aprendiz através do resultado do questionário, usando uma escala de classificação e obtendo seu perfil.
2	Com o perfil definido, o sistema usa uma tabela de pares estado-ação. Nesta, as ações representam estratégias pedagógicas adotadas pelo tutor para guiar o aprendiz.
3	O sistema escolhe a ação como maior valor de utilidade baseado nos resultados do algoritmo *Q-Learning*.
4	O tutor executa a ação junto ao aprendiz e obtém resultados.
5	Estes resultados são encaminhados para o Módulo Diagnóstico, que calcula reforços positivos ou negativos para o par estado-ação $r(s_t, a)$ e atualiza na tabela o valor de utilidade $Q_t(s_t, a_t)$.
6	Depois de atualizado o valor de utilidade na tabela, o sistema volta a reclassificar o aprendiz na escala e retorna ao passo 2.

Algoritmo 2 - Funcionamento do Módulo de Aprendizagem por Reforço para Modelagem Autônoma do Aprendiz.

2.3 Metaheurísticas

Segundo Bianchi e Costa (2005), uma heurística pode ser definida como uma técnica que melhora, no caso médio, a eficiência na solução de um problema. “Funções heurísticas são a forma mais comum de se aplicar o conhecimento adicional do problema a um algoritmo de busca” (NORVING; RUSSEL, 2013), sendo, dessa forma, uma maneira de generalização do conhecimento que se tem acerca de um domínio.

Glover e Kochemberger (2003) defendem que metaheurísticas são métodos de solução que coordenam procedimentos de busca locais com estratégias de mais alto nível, de modo a criar um processo capaz de escapar de mínimos locais e realizar uma busca robusta no espaço de soluções de um problema.

2.3.1 Busca Tabu

A Busca Tabu é uma metaheurística proposta por Glover (1986) e, posteriormente, detalhada em Glover e Laguna (1997). Tem como característica a construção de vizinhanças de possíveis soluções através de uma rotina iterativa que proíbe o bloqueio em um ótimo local.

De uma solução inicial, um algoritmo BuscaTabu explora a cada iteração um conjunto de vizinhos da solução. O vizinho da solução corrente com melhor avaliação se torna

a nova solução, mesmo que tenha uma menor avaliação. A Figura 6 a seguir apresenta um esboço da evolução do processo de geração de soluções vizinhas. O melhor vizinho representado por S_i^* é tomado a cada iteração como a solução corrente.

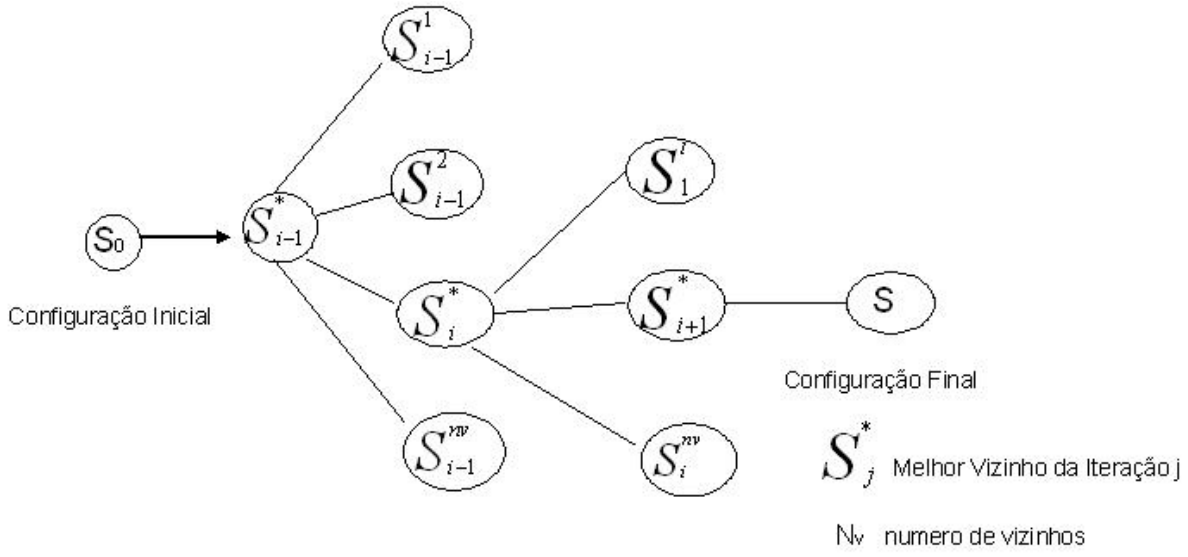

Figura 6 - Esquema de geração de vizinhança para Busca Tabu (GLOVER; LAGUNA, 1997).

Essa estratégia de tomar o melhor vizinho como a nova solução é utilizada para escapar de mínimos locais, porém pode fazer com que o algoritmo forme ciclos, ou seja, retorne a uma solução já tomada anteriormente passando pelo mesmo caminho. Como, por exemplo, na Figura 6, se for considerado que S_i^* é a solução corrente, a solução da iteração anterior representada por S_{i-1} também pertencerá à vizinhança da solução atual. Desse modo, caso seja a melhor solução da vizinhança, será tomada novamente como a solução corrente.

A palavra Tabu teve origem na ilha de Tonga da Polinésia, e de modo geral, significa que um comportamento ou assunto é proibido ou sagrado. A característica mais importante e semelhante ao seu significado original vem do conceito que Tabus são concebidos de uma memória social de assuntos proibidos que esteja sujeita a modificação com o passar do tempo. Desse modo, para evitar a ocorrência de ciclos, existe uma lista de movimentos proibidos, denominado lista Tabu. Na sua forma mais clássica contém os últimos movimentos realizados em uma fila de tamanho fixo, ao qual o primeiro elemento que entra é o primeiro que sai. Assim, na Busca Tabu ficam excluídos os vizinhos que estão na lista tabu, mesmo que estes sejam boas soluções da vizinhança atual.

2.3.2 GRASP

A *Greedy Randomized Adaptive Search Procedure* (GRASP) é uma metaheurística que foi proposta por Feo e Resende (1995). O funcionamento dessa metaheurística consiste de duas fases: uma fase de construção, na qual uma solução é gerada, elemento a elemento; uma fase de busca local, onde um ótimo local na vizinhança da solução construída é pesquisado.

Na metaheurística GRASP (SANTOS, 2009) a fase de construção é incremental. A cada iteração dessa fase, um conjunto de elementos candidatos é formado por todos os elementos que ainda não foram incorporados à solução parcial em construção sem que, com isso, comprometa a sua viabilidade. Para a seleção do próximo elemento a ser incorporado, realiza-se a avaliação de todos os elementos candidatos, estabelecendo uma lista ordenada dos elementos. A avaliação está associada ao aumento ou diminuição incremental na função objetivo devido à incorporação deste elemento na solução em construção. Isto conduz à criação de uma lista restrita de candidatos (LRC) cujo número de elementos é parametrizado pelo algoritmo. Na escolha de um elemento da LRC, utiliza-se um critério aleatório. O ordenamento guloso (do elemento de maior para o de menor valor), a adaptabilidade do tamanho da LRC, e o critério de escolha aleatório de seus elementos são diretamente representados na denominação da metaheurística.

Um dos pontos mais importantes na heurística GRASP é a escolha do comprimento da LRC, representado pelo parâmetro α_{LRC}. A influência desta escolha sobre a qualidade da solução construída é estudada em Ribeiro e Resende (2003). Após a construção da solução inicial, utiliza-se um procedimento de busca local para a melhoria da solução, dentro de uma vizinhança especificada. Existem várias formas de definição desta vizinhança, dependendo do problema de otimização que se pretende resolver. Para o problema do caixeiro viajante, normalmente são utilizados critérios baseados em trocas, tal como 2 - opt e 3 - opt: partindo-se da solução corrente, a cada etapa de busca local analisam-se todas as combinações decorrentes da troca de posição entre duas (ou três) cidades, escolhendo-se a solução que produz o melhor valor da função objetivo (JHONSON; MCGEOCH, 2003).

O Algoritmo 3 apresenta o padrão para a metaheurística GRASP, em que se tem a fase da construção da solução inicial aleatória gulosa e logo após a busca local, na qual se tenta melhorar a solução inicial. Finalmente, tem-se o armazenamento da melhor solução.

```
1  procedimento GRASP(maxIterações, α_LRC)
2  CarregaInstância();
3  para i=0 to maxIterações faça
4     solução ← ConstruçãoGulosaAleatória(α_LRC)
5     solução ← BuscaLocal (solução, Vizinhança(solução))
6     AtualizaSolução(solução, melhorSolução)
7  fim para
8  retorne melhorSolução
```

Algoritmo 3: Metaheurística GRASP (SANTOS, 2009).

No Algoritmo 4 tem-se a fase de construção do GRASP, em que a solução inicial é desenvolvida de forma incremental, a partir da lista restrita de candidatos.

```
1  procedimento ConstruçãoGulosaAleatória(α_LRC)
2  solução ← 0
3  enquanto solução não completa faça
4     ConstruirLRC (LRC)
5     s ← SelecionaElementoAleatório(LRC, α_LRC)
6     solução ← solução + s
7  fim enquanto
8  retorne solução
```

Algoritmo 4: Fase de construção da metaheurística GRASP (SANTOS, 2009).

No Algoritmo 5 tem-se a fase de busca local para melhoria da solução inicial em sua vizinhança.

```
1  procedimento BuscaLocal(solução, VizinhançaSolução(solução))
2  solução ← 0
3  enquanto solução local não for ótimo faça
4     Encontrar melhor solução s ∈ Vizinhança(solução) solução ← s
5  fim enquanto
6  retorne solução
```

Algoritmo 5: Busca local da metaheurística GRASP (SANTOS, 2009).

2.4 Trabalhos Correlatos

Diante da dificuldade em se obter o modelo do estado cognitivo do aprendiz, Guelpeli, Omar e Ribeiro (2004) propõem alterações na arquitetura clássica dos STIs. Nesse

trabalho, é adicionado um novo módulo de diagnóstico onde são aplicadas técnicas de AR, através do algoritmo *Q-Learning* (WATKINS, 1986), o que possibilita modelar autonomamente o aprendiz. Um valor de utilidade é calculado baseado em uma tabela de pares estado-ação, a partir da qual o algoritmo estima reforços futuros que representam os estados cognitivos do aprendiz. A melhor política a ser usada pelo tutor para qualquer estado cognitivo do aprendiz é disponibilizada pelo próprio algoritmo de AR, sem que seja necessário um modelo explícito do aprendiz. A fase de exploração do algoritmo *Q-Learning* é realizada através da herística aleatória, ou seja, a cada iteração a ação é escolhida aleatoriamente. A fase de explotação utiliza a herística gulosa, ou seja, a ação escolhida a cada iteração é aquela que possui o maior valor de qualidade na tabela de pares estado-ação.

O emprego de metaheurísticas com objetivo de acelerar o aprendizado por reforço do algoritmo *Q-Learning* é observado em Bianchi e Costa (2005). Nesse trabalho, uma nova classe de algoritmos é proposta. Denominada de "Aprendizado Acelerado por Heurísticas", nessa classe de algoritmos a heurística é usada somente para a escolha da ação a ser tomada, não modificando o funcionamento do algoritmo de AR, preservando muitas de suas propriedades. Foram propostos cinco algoritmos e testados em diversos domínios, concluindo que mesmo uma heurística muito simples resulta em aumento significativo do desempenho do algoritmo de AR utilizado. No entanto, o domínio de Sistemas Tutores Inteligentes não foi objeto de estudo.

A metaheurística Busca Tabu com o objetivo de balancear a exploração e a explotação é apresentada em Zhang e Liu (2008). Nesse trabalho, introduze-se o pensamento da Busca Tabu no funcionamento do algoritmo *Q-Learning*. Batizado de T-Q-Learning, a taxa de convergência desse algoritmo se mostra mais rápida e evita soluções parcialmente ótimas em seus experimentos. O domínio objeto desse estudo foi simulação de subida de encosta.

No domínio de sistemas tutores inteligentes, Javadi, Masoumi e Meybodi (2012) propõem um método para melhorar o comportamento do modelo do aprendiz. Nesse método proposto, o modelo do aprendiz é determinado por autômatos de alto nível, denominados Level Determinant Agent (LDA) que tentam caracterizar e promover o modelo de aprendizagem dos estudantes. LDA usa autômatos de aprendizagem como mecanismo de aprendizagem para mostrar o quanto o aluno é lento, normal ou rápido em termos de aprendizagem. Nesse trabalho a aprendizagem por reforço também é realizada através do algoritmo *Q-Learning*, no entanto não são empregadas metaheurísticas com o objetivo de acelerar essa aprendizagem.

A introdução de técnicas de inteligência artificial pode ser aplicada em ambientes virtuais de aprendizagem tradicionais. Palomino (2013) elaborou um modelo de ambiente

inteligente de aprendizagem, inspirado no funcionamento de tutores inteligentes, baseado em agentes, para dar adaptabilidade a ambientes virtuais de aprendizagem distribuídos. O AVA Moodle foi utilizado como estudo de caso, levando em conta o desempenho do aluno nas tarefas e atividades propostas pelo professor e acompanhando o acesso dele ao material de estudo. Nesse trabalho, foram implementados os agentes para, através do envio de mensagens aos alunos e da configuração dos recursos e atividades a serem disponibilizados na disciplina, procederem as ações pedagógicas definidas pelo professor, de forma individualizada e adaptativa a cada aluno, em cada disciplina. O algoritmo *Q-Learning* não é empregado nesse trabalho bem como metaheurísticas.

Sistemas Tutores Inteligentes podem ser empregados em diversos domínios, inclusive no ensino de idiomas. Deephi e Sasikumar (2014) descrevem o modelo do aprendiz para um Sistema Tutor de Idiomas Inteligente com o objetivo de ensinar as regras gramaticais de inflexão verbal/nominal da língua Telugu. Neste trabalho, o modelo do aprendiz é utilizado para representar o grau de confiança de um estudante aplicando cada regra gramatical, que é utilizada pelo tutor para gerar problemas o aprimorando em áreas fracas. Um estudo piloto é realizado para medir a eficácia do sistema.

Outro emprego dos Sistemas Tutores Inteligentes é a aquisição automatizada de conhecimento (AAC). Li e Zhou (2015) apresentam a arquitetura de um sistema com esse objetivo, capaz de melhorar a eficiência de ensino. Esse sistema realiza uma análise baseada no desempenho dos estudantes e seleciona os materiais de estudo de maneira apropriada ao nível de conhecimento do estudante e conhecimento prévio.

A Tabela 1 apresenta os trabalhos correlatos citados, para cada um deles é apresentado o emprego ou não das áreas de Sistemas Tutores Inteligentes, Aprendizagem Autônoma, Algoritmo *Q-Learning* e Utilização de Metaheurísticas:

Tabela 1 - Trabalhos correlatos

Autor, Ano	**Sistemas Tutores Inteligentes**	**Aprendizagem Autônoma**	**Algoritmo *Q-Learning***	**Utilização de metaheurísticas**
GUELPELI, OMAR e RIBEIRO, 2004	X	X	X	
BIANCHI e COSTA, 2005			X	X
ZHANG e LIU, 2008			X	X
JAVADI, MASOUMI e MEYBODI, 2012	X	X	X	
PALOMINO, 2013	X	X		
DEEPHI e SASIKUMAR, 2014	X			
LI e ZHOU, 2015	X			

3 ACELERAÇÃO DE CONVERGÊNCIA ATRAVÉS DE META-HEURÍSTICAS

Neste capítulo serão apresentados os procedimentos metodológicos adotados para a execução das simulações presentes nesse trabalho. Será detalhado o funcionamento das metaheurísticas Lista Tabu e GRASP aplicadas na fase exploratória do algoritmo *Q-Learning*.

3.1 Descrição do Modelo

Na simulação apresentada nesse trabalho o algoritmo *Q-Learning* (WATKINS, 1989) é utilizado para realizar o Aprendizado por Reforço. Nesse algoritmo, a escolha de uma ação é baseada em uma função de utilidade que mapeia estados e ações a um valor numérico. Em Guelpeli, Omar e Ribeiro (2004), o valor de utilidade $Q(s, a)$ de um par (estado(s), ação(a)), é calculado a partir de reforços medidos pela qualidade do estado cognitivo do aprendiz. Portanto, o principal objetivo do algoritmo *Q-Learning* é estimar autonomamente, em cada estado que o aprendiz se encontra, a ação com maior valor de utilidade. Como consequência, conseguirá estimar o estado cognitivo do aprendiz ($estado_{t+1}$). Na simulação, a ação representa uma estratégia escolhida pelo tutor para ensinar um determinado conteúdo para o aprendiz como, por exemplo, escolher entre aula expositiva, vídeo aula, fichamento ou estudo de caso.

A política pedagógica representa o grau de rigor ao se avaliar o desempenho do aprendiz em aprender um determinado conteúdo. Existem instituições de ensino, por exemplo, que consideram um aluno com aprendizado satisfatório para aprovação quando alcança o desempenho de 50% em avaliações, outras, no entanto, 60 ou 70%. Numa política pedagógica mais restritiva, o rigor dessa avaliação é maior em relação a uma política pedagógica menos restritiva. O protótipo usado na simulação não conhece os modelos obtidos, então a convergência se dá através do aprendizado da política ótima de ações, ou seja, o que foi determinado na política pedagógica. Os experimentos foram realizados sobre um protótipo de tutor, em um ambiente com uma matriz 5x10 mapeando estados e ações (5 estados e 10 ações) com elementos descritos a seguir.

- Um conjunto de estados

$S = \{E0, E1, E2, E3, E4\}$,

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 i. Os intervalos numéricos representam o número de acertos do aprendiz em um questionário formado por 10 questões. Com isso, estima-se que esse aprendiz tenha um grau cognitivo baseado nesses estados, onde:

$E0 \Rightarrow [0, 2]$,

$E1 \Rightarrow]2, 4]$,

$E2 \Rightarrow]4, 6]$,

$E3 \Rightarrow]6, 8]$,

$E4 \Rightarrow]8, 10]$.

ou seja, caso o aprendiz tenha obtido o desempenho de 6 acertos será classificado como estado $E2$.

- Um conjunto de ações

$A = \{A0, A1, A2, A3, A4, A5, A6, A7, A8, A9\}$

que podem ser escolhidas pelo tutor. Cada ação pode corresponder à aplicação de provas, exercícios, questionários, perguntas, trabalhos, testes, outras ações pedagógicas, ou combinações destes e outros dispositivos de avaliação, usados pelo tutor, seguindo as estratégias pedagógicas estabelecidas.

- Um conjunto de Reforços Instantâneos associados a cada estado visitado, ou seja,

$E0 \Rightarrow r = 1 - Ruim$;

$E1 \Rightarrow r = 3 - Regular$;

$E2 \Rightarrow r = 5 - Bom$;

$E3 \Rightarrow r = 7 - Muito\ Bom$;

$E4 \Rightarrow r = 10 - Excelente$.

Foi definida uma metodologia de teste na qual foram criados três modelos determinísticos e não determinísticos: **M1** representando o modelo de aprendiz Ruim, **M2** aprendiz **Bom** e **M3** aprendiz **Excelente**. Foram criadas duas políticas pedagógicas **P1** e **P2**, onde **P2** é uma política mais restritiva do que **P1** em relação ao modelo, já que os intervalos entre as ações são menores, possibilita desta forma uma depuração maior das ações em relação aos estados, sendo assim haverá um número maior de decisões para cada estado com o uso da política **P2**. Para as simulações apresentadas nesse trabalho, foram adotados os modelos não determinísticos (**M1**, **M2** e **M3**) e a política pedagógica **P1**.

O trabalho presente em Guelpeli et al. (2012) cria simulações variando os valores de α (Taxa de Aprendizagem) e γ (Desconto temporal), parâmetros encontrados no algoritmo *Q-Learning*, analisando a influência dessa variação e o tempo de convergência do algoritmo. Esse trabalho conclui que a variação desses parâmetros interfere no aprendizado do STI e, consequentemente, no diagnóstico do modelo do aprendiz. Os valores adotados para as simulações apresentadas neste trabalho desses parâmetros são α = 0,9 e γ = 0,9; valores que obtiveram os melhores resultados em Guelpeli et al. (2012), ou seja, maior valor de utilidade das ações e reforço.

O aprendizado do valor de utilidade de cada uma das ações é obtido autonomamente pelo algoritmo *Q-Learning* através da interação do agente com o ambiente, nesse caso, tutor e aprendiz. Essa interação pode ser visualizada na Figura 7.

Figura 7 - Interação entre o agente de aprendizagem e o ambiente (BELLMAN, 1957).

O algoritmo *Q-Learning* é dividido em duas fases: fase de exploração e fase de explotação. A fase de exploração do algoritmo objetiva explorar ações desconhecidas ou pouco visitadas. Essa fase permite conhecer quais ações maximizam o valor das recompensas obtidas no tempo. De posse desse conhecimento, a fase de explotação do algoritmo busca atingir essa maximização utilizando as ações que obtiveram melhor desempenho durante a fase de

exploração. As diferentes estratégias para que o algoritmo alcance o objetivo da fase está definida em cada uma das metaheurísticas.

O passo a passo do funcionamento da utilização de metaheurísticas no módulo de diagnóstico do sistema tutor inteligente é representado no Algoritmo 6:

1	O tutor interage com o aprendiz através da aplicação de uma política pedagógica ($ação_t$)
2	Essa interação altera o estado cognitivo do aprendiz ($estado_{t+1}$)
3	Um novo valor de reforço é calculado ($reforço_{t+1}$) de acordo com a qualidade da ação escolhida
4	O módulo de diagnóstico utiliza o algoritmo *Q-Learning* para o cálculo das melhores ações possíveis de acordo com um determinado estado cognitivo do aprendiz
5	Caso esteja na fase de exploração: seleciona a melhor ação através da metaheurística de exploração. Caso esteja na fase de explotação: seleciona a melhor ação através da metaheurística de explotação
6	O aprendiz interage novamente com o tutor

Algoritmo 6 – Introdução de metaheurísticas nas fases de exploração e explotação do algoritmo *Q-Learning*

O modelo proposto neste trabalho está representado na Figura 8:

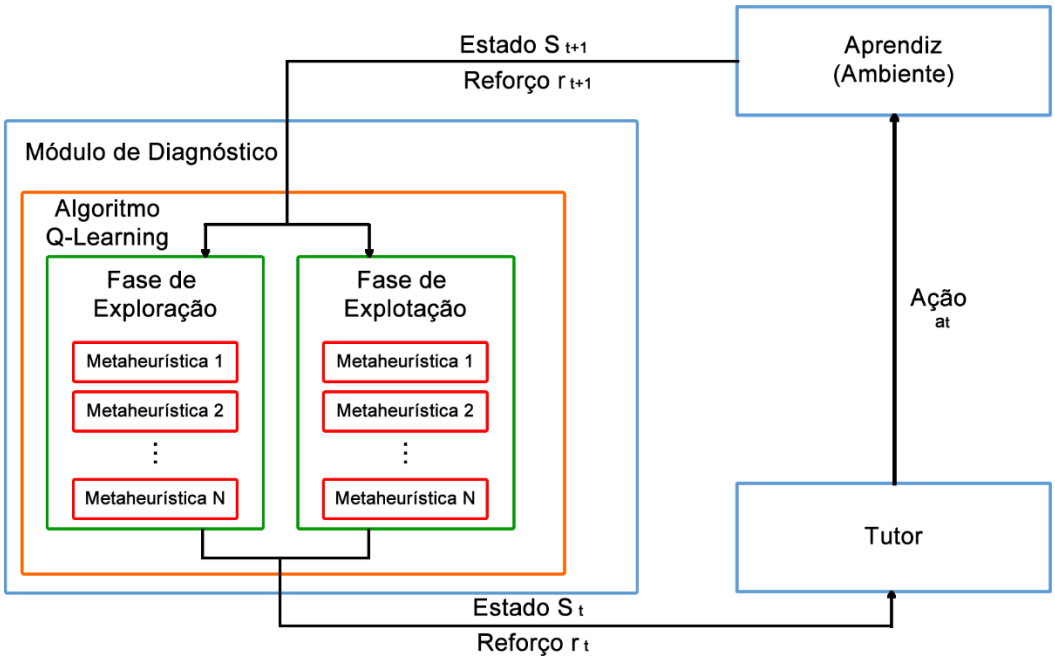

Figura 8 – Aceleração da convergência do algoritmo *Q-Learning* através de metaheurísticas

As metaheurísticas Lista Tabu e GRASP foram inseridas na fase exploratória desse algoritmo e seu funcionamento será detalhado nas próximas seções.

3.2 Exploração com a Metaheurística Lista Tabu

Nesta seção serão apresentadas as características de funcionamento da exploração baseada na metaheurística Lista Tabu.

3.2.1 Ativação da Lista Tabu

Com a finalidade de garantir uma exploração mínima de pelo menos uma visita a cada uma das ações A para cada um dos estados S, a Lista Tabu de cada estado só é ativada ao término dessa exploração inicial.

3.2.2 Critério de Aspiração

Na metaheurística Busca Tabu quando uma solução definida como proibida é escolhida como a melhor solução, ela poderá ser admitida caso atenda ao Critério de Aspiração. Usualmente o critério de aspiração perdoa a solução proibida caso seu resultado seja melhor que o melhor resultado encontrado até agora.

Neste trabalho, a função objetivo ($max_a Q^u_{t\ (s_{t+1},a)}$) trabalha com os valores na tabela de valores de utilidade das ações utilidade $Q(s,a)$ obtidos nas iterações anteriores. O critério de aspiração adotado é o conceito de "melhor resultado até agora", proposto em Zhang e Liu (2008). Caso a solução inicial encontrada seja proibida, ou seja, pertença à lista Tabu, o algoritmo testará se o critério de aspiração é atendido. Para atender ao critério de aspiração a solução encontrada precisar ser o maior valor de $Q(s,a)$ já encontrado nesse estado s do aprendiz, nesse caso a solução atende ao critério e é "perdoada", portanto poderá ser utilizada. Foi criada uma matriz de tamanho 5 para armazenar o melhor resultado já encontrado de $Q(s,a)$ para cada um dos estados (E0, E1, E2, E4 e E5). Sempre que uma solução adotada em um determinado estado supera esse valor, o respectivo valor na matriz é atualizado.

A Figura 9 apresenta um exemplo de funcionamento da metaheurística Lista Tabu. Nesse exemplo o tamanho da Lista Tabu adotado é de dois elementos, o estado cognitivo atual do aprendiz representado é E2 (linha E2 da tabela), as soluções A1 e A3 foram adicionadas à Lista Tabu. Nesse caso, se a solução A3 atender ao critério de aspiração, sou seja, o valor 38,79 é o maior valor já encontrado para o estado E2, esta solução atende ao critério de aspiração e está "perdoada" para ser utilizada. Caso contrário, a solução adotada será A0, maior valor dentre as soluções não proibidas (fora da Lista Tabu).

	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9
E0	37,15	0	0	0	0	0	0	0	0	0
E1	0	0	0	0	0	0	17,68	0	0	0
E2	11,49	34,2	9,82	38,79	4,5	10,37	4,5	5,83	4,5	4,5
E3	0	0	37,38	0	0	17,49	0	0	0	0
E4	0	0	0	0	0	0	0	0	0	0

Legenda
Solução Tabu

Figura 9 - Funcionamento da metaheurística Lista Tabu

O tamanho da Lista Tabu define o número de soluções que, adotadas, serão proibidas enquanto permanecerem na Lista Tabu. Foram realizados experimentos com diversos valores de tamanho da Lista Tabu com o objetivo de testar qual tamanho alcançaria os maiores valores de $Q(s,a)$. A Figura 10 apresenta o desempenho da Lista Tabu com os tamanhos 2, 3, 5 e 7 elementos para o modelo **M2** – **Bom**, **500** passos:

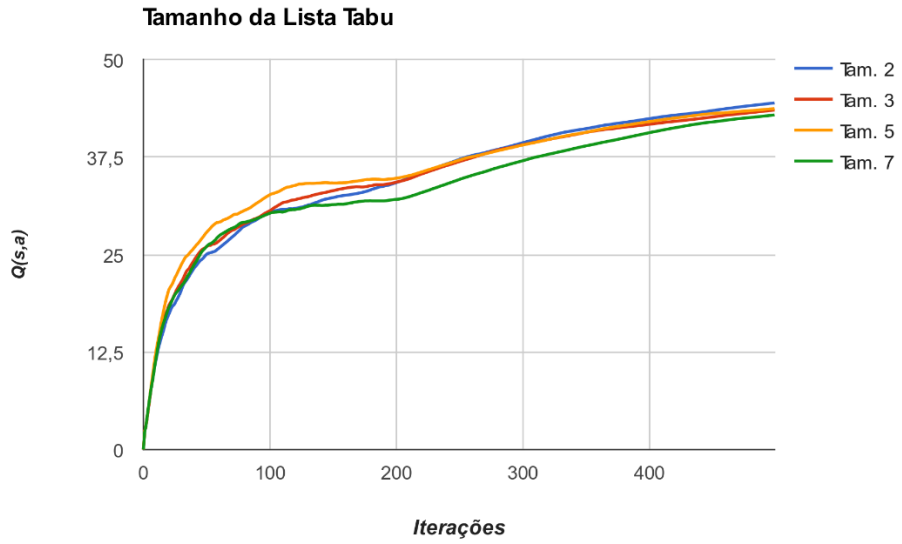

Figura 10 - Comparação de desempenho dos tamanhos da lista tabu – Modelo M2 - Bom - 500 passos.

Apesar do desempenho semelhante, a Lista Tabu de tamanho 2 elementos apresentou os melhores resultados de $Q(s,a)$ e será adotada como tamanho para as simulações comparativas entre as metaheurísticas.

3.3 Exploração com a Metaheurística GRASP

Nesta seção serão apresentadas as características de funcionamento da exploração baseada na metaheurística GRASP.

3.3.1 Tamanho da Lista Restrita de Candidatos

O tamanho da Lista Restrita de Candidatos define o número de soluções iniciais que serão geradas. Entre essas soluções, uma solução é adotada e sua escolha é aleatória. Foram realizados experimentos com diversos valores de tamanho de tamanho da Lista Restrita de Candidatos com o objetivo de testar qual tamanho alcançaria os maiores valores de $Q(s,a)$. A Figura 11 apresenta o desempenho da metaheurística GRASP com a Lista Restrita de Candidatos com os tamanhos 2, 3, 4 e 5 elementos para o modelo **M2** – **Bom**, **500** passos:

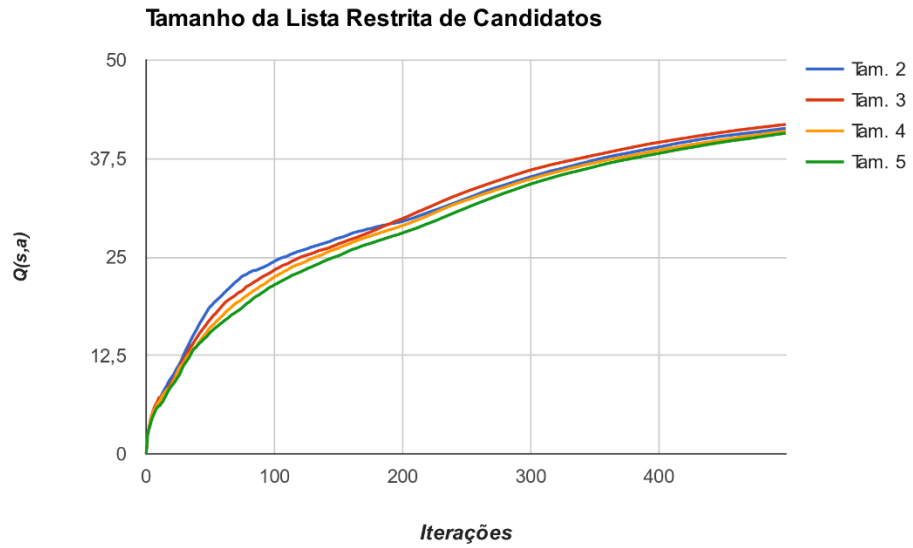

Figura 11 - Comparação de desempenho dos tamanhos da Lista Restrita de Candidatos – Modelo M2 - Bom - 500 passos.

Apesar do desempenho semelhante, a Lista Restrita de Candidatos de tamanho 3 elementos apresentou os melhores resultados de $Q(s,a)$ e será adotada como tamanho para as simulações comparativas entre as metaheurísticas.

A Figura 12 apresenta um exemplo de funcionamento da metaheurística GRASP. Nesse exemplo o tamanho da Lista Restrita de Candidatos é de três elementos, o estado cognitivo atual do aprendiz representado é E2 (linha E2 da tabela), as soluções A0, A1 e A3 foram as soluções adicionadas na lista. Nesse caso, uma dessas soluções será escolhida aleatoriamente como a solução adotada pela metaheurística. Essa estratégia tem por objetivo evitar que o algoritmo fique preso em ótimos locais, permitindo que soluções promissoras também sejam escolhidas.

	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9
E0	37,15	0	0	0	0	0	0	0	0	0
E1	0	0	0	0	0	0	17,68	0	0	0
E2	11,49	34,2	9,82	38,79	4,5	10,37	4,5	5,83	4,5	4,5
E3	0	0	37,38	0	0	17,49	0	0	0	0
E4	0	0	0	0	0	0	0	0	0	0

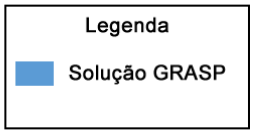

Figura 12 - Funcionamento da metaheurística GRASP

4 SIMULADOR

Neste capítulo será apresentado o simulador desenvolvido para os experimentos realizados neste trabalho. Serão exibidas as telas do programa, as configurações disponíveis para as simulações, o acompanhamento dos resultados parciais das simulações e os arquivos de saída com os resultados.

4.1 Interface

A interface do simulador é dividida em 3 telas: Configuração, Simulação e Visita aos Estados. A tela Configuração apresenta as configurações disponíveis de funcionamento das simulações. A tela Simulação apresenta os resultados parciais em tempo real das simulações em execução. A tela Visita aos Estados apresenta gráficos de pizza com o resultado do percentual de visita aos estados de cada configuração da simulação. As telas do simulador são apresentadas no Apêndice A.

4.2 Configurações da simulação

Quanto ao aprendiz, é possível selecionar entre os modelos **Ruim**, **Bom** e **Excelente**. O modelo de aprendiz representa a capacidade do aprendiz de assimilar um conteúdo ao recebe-lo através de uma estratégia pedagógica. O modelo **Ruim** representa menor capacidade e o modelo **Excelente**, maior.

Quanto aos tipos de modelo, é possível selecionar modelo **Determinístico** e **Não Determinístico**. No modelo determinístico o aprendiz, ao receber um determinado conteúdo através de uma política pedagógica, sempre produzirá um mesmo número de acertos e erros. No modelo não determinístico a quantidade de acertos e erros estará associada a um comportamento probabilístico.

Quanto à Política Pedagógica do Tutor é possível selecionar a política **P1** e **P2**. A política pedagógica representa o rigor do tutor ao avaliar as respostas do aprendiz em um questionário a respeito do conteúdo apresentado ao aprendiz. A política **P2** é mais restritiva em relação a política **P1**.

A Figura 13 apresenta a interface do simulador para a seleção das configurações de Modelo de Aprendiz, Tipo de Modelos e Política Pedagógica do Tutor:

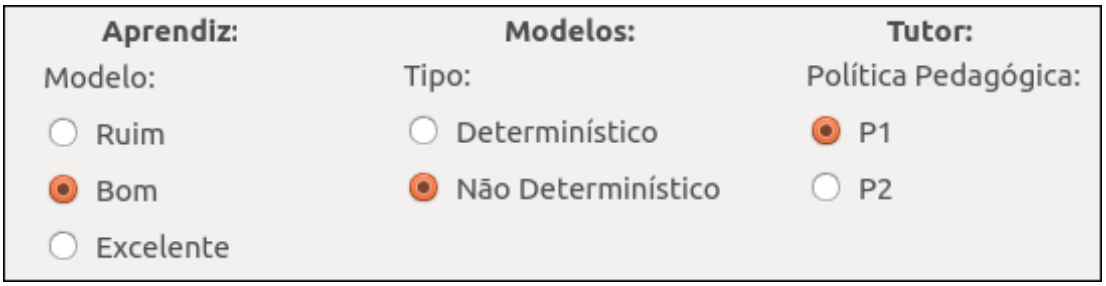

Figura 13 – Simulador – Configuração Aprendiz, Modelo e Tutor

Nas configurações de simulação é possível definir as seguintes opções: **N° Iterações**, **N° de simulações**, **Atraso**, **Tamanho min Tabu**, **Tamanho lista RC**. O parâmetro **N° de Iterações** define quantos questionários serão aplicados ao aprendiz em cada simulação. O parâmetro **N° de simulações** define quantas simulações serão realizadas para cada configuração de funcionamento do algoritmo *Q-Learning*. Ao final de todas as simulações, os resultados são calculados através de média entre essas simulações. O parâmetro **Atraso**, informado em milissegundos, controla o intervalo de execução entre a aplicação de um questionário e outro. O valor mínimo definido para esse parâmetro é de 50 milissegundos com o objetivo de permitir a atualização de todos os componentes gráficos do simulador com os valores atuais da simulação (barras de progresso, rótulos, gráficos, etc). O parâmetro **Tamanho min Tabu** define do tamanho da lista Tabu, ou seja, quantas das últimas soluções adotadas ficarão bloqueadas na lista Tabu. O parâmetro **Tamanho lista RC** define quantas melhores soluções candidatas serão selecionados pelo STI para posterior escolha aleatória. No parâmetro **Automática** o intervalo de aplicação entre um questionário e outro é automático, respeitando o intervalo definido no parâmetro **Atraso**. No parâmetro **Manual**, para prosseguir para a próxima aplicação de questionário é necessário que o usuário clique no botão **Próxima** ao final de cada aplicação do formulário. A Figura 14 apresenta os parâmetros de configuração das simulações:

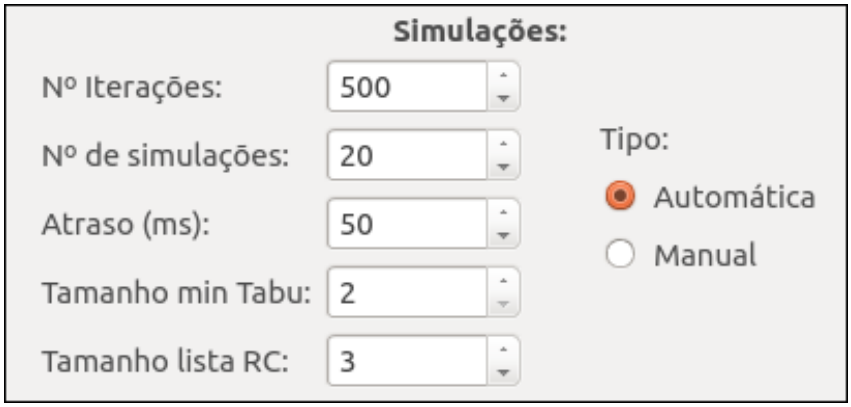

Figura 14 – Simulador – Configuração das Simulações

A execução do algoritmo *Q-Learning* é dividida em 2 fases: fase de exploração e fase de explotação. Nas configurações do simulador é possível selecionar as heurísticas que serão utilizadas como **Estratégias de Exploração** e **Estratégias de Explotação** durante as simulações. Cada par estratégia de exploração – estratégia de explotação forma uma configuração de simulação para gerar resultados comparativos. A Figura 15 apresenta as estratégios disponíveis no STI:

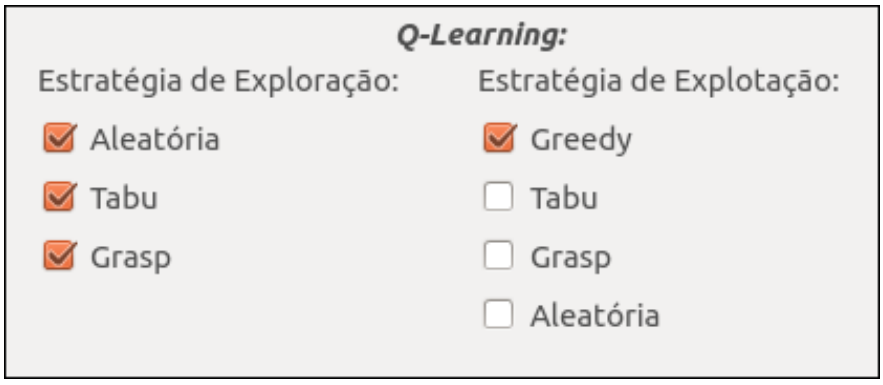

Figura 15 – Simulador – Configuração do Algoritmo *Q-Learning*

4.3 Acompanhamento das simulações

Quando uma simulação é iniciada, é possível acompanhar os resultados parciais de cada passo da execução através da tela Simulação (Apêndica A). O campo **Iteração** apresenta o passo atual da simulação, ou seja, em ordem cronológica quantos questionários foram aplicados pelo tutor ao aprendiz. O campo **Reforço** informa o valor de reforço calculado pelo algoritmo *Q-Learning* ao aplicar a última estratégia pedagógica. O campo **Ação** exibe qual a última estratégia pedagógica adotada. O campo **Acertos** informa o número de acertos e **Erros** o número de erros do último questionário aplicado pelo tutor ao aprendiz. O campo **Estado Cognitivo Atual** informa o estado cognitivo atual do aprendiz e **Estado Cognitivo Futuro**

informa para qual estado a política pedagógica conseguiu levar o aprendiz, ou seja, o estado cognitivo do aprendiz no próximo passo da simulação. O campo **Tam. Listas Tabu** exibe o tamanho atual das estruturas de controle de tamanho da lista Tabu (**T0**, **T1**, **T2**, **T3** e **T4**). Os campos **Exploração** e **Explotação** informam em que fase o algoritmo *Q-Learning* se encontra. Os campos **Estratégia de Exploração** e **Estratégia de Explotação** informam qual configuração de simulação atual a execução da simulação se encontra. A Figura 16 apresenta os campos para acompanhamento dos resultados parciais da simulação:

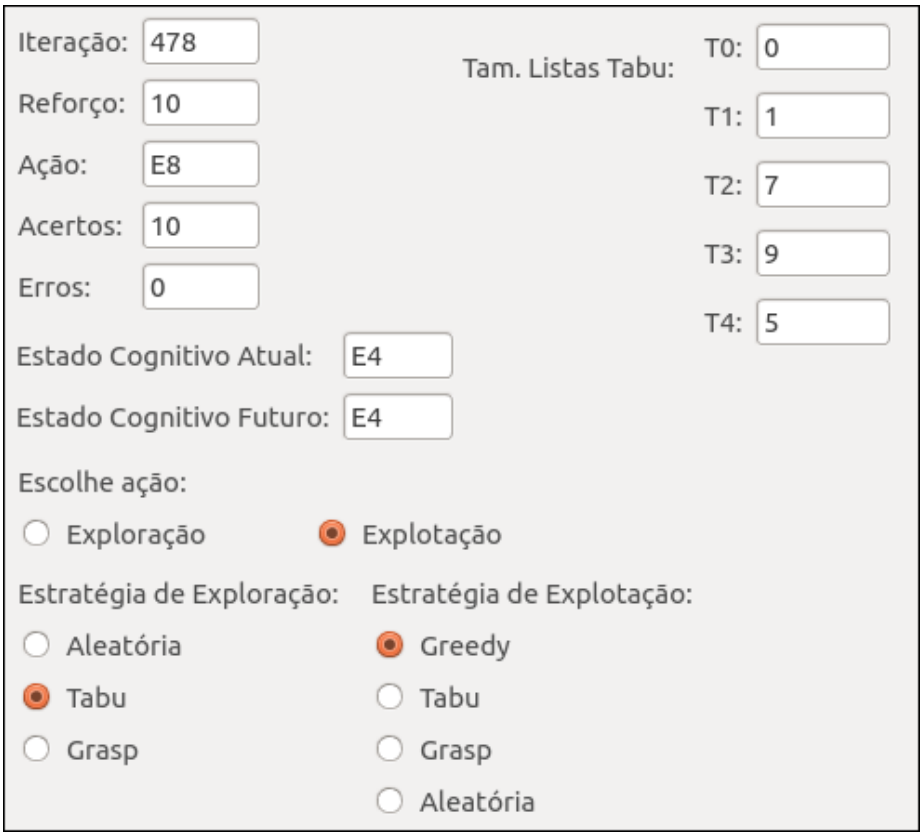

Figura 16 – Simulador – Acompanhamento da Simulação – Resultados parciais

Através da tabela $Q(s,a)$ é possível acompanhar o mapeamento do valor de qualidade de cada uma das ações (colunas $A0$ à $A9$) em um determinado estado cognitivo do aprendiz (linhas $E0$ à $E4$). Conforme um determinado par estado-ação é visitado com maior frequência, o STI intensifica a tonalidade da cor de fundo da célula na cor vermelho a afim de destacar essa situação. A Figura 17 apresenta a visualização da tabela $Q(s,a)$, note que neste exemplo a ação A6 do estado cognitivo E4 foi adotada mais vezes que as demais ações (maior intensidade na cor de fundo vermelho):

	A0	A1	A2	A3	A4	A8	A6	A7	A8	A9
E0	6.3	0	0	0	0	0	0	0	0	0
E1	0	0	55.9846	0	0	0	0	0	0	0
E2	9.07681	0	6.3	0	18.4273	6.3	60.8478	0	9	11.8345
E3	0	7.24598	4.5	6.88077	7.60499	5.9972	13.0439	11.59	88.6004	9
E4	7.10494	35.8186	6.61472	19.1075	26.0006	9.42701	41.2201	9	79.7844	11.3979

Figura 17 – Simulador – Acompanhamento da Simulação – Tabela $Q(s,a)$

Os valores obtidos de **Reforço**, **Transição de Estados** e $\boldsymbol{Q(s,a)}$ podem ser acompanhados através de gráficos, conforme Figura 18. Cada configuração de simulação é destacada em uma determinada cor e, ao final das simulações, a média dos resultados obtidos é apreentada possibilitando uma comparação visual.

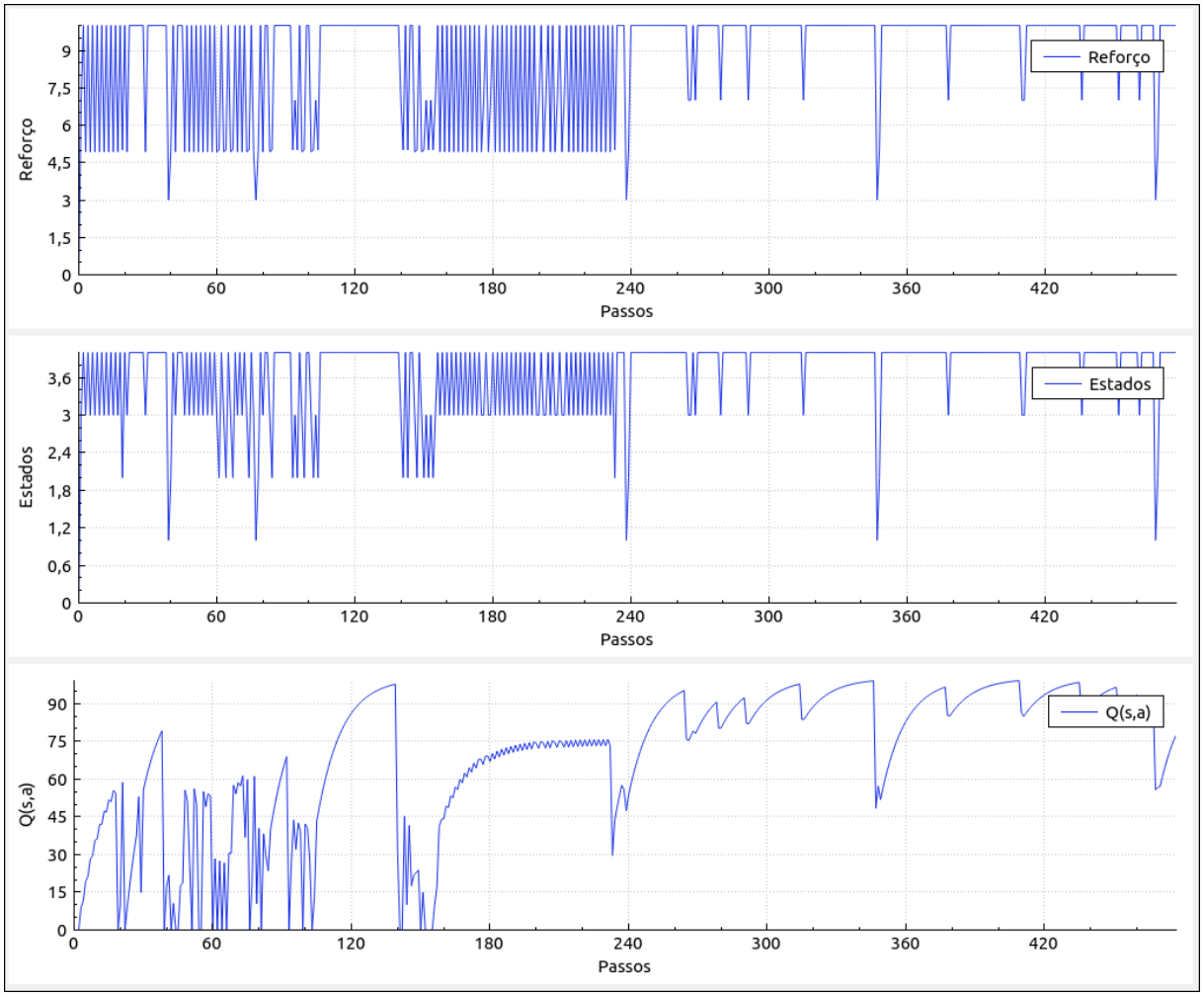

Figura 18 – Simulador – Acompanhamento das Simulações – Reforço, Estados e $Q(s,a)$

A tela Visita aos Estados (Apêndice A) apresenta gráficos de pizza com o percentual de visitas de a cada um dos estados cognitivos do aprendiz para cada uma das configurações das simulações. Ao final das simulações, o resultado médio do percentual de visitas é apresentado, conforme Figura 19:

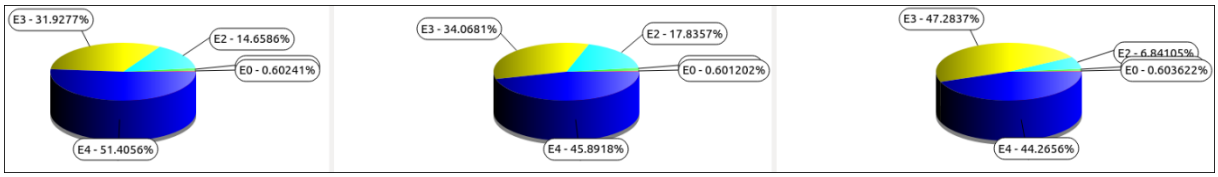

Figura 19 – Simulador – Acompanhamento das Simulações – Visita aos Estados

4.4 Arquivos de Saída

O simulador grava os resultados parciais e finais das simulações em arquivos no formato CSV. Dessa maneira é possível produzir planilhas e gráficos comparativos entre as configurações de simulação. Cada arquivo de saída tem o seguinte padrão de nomenclatura:

- Modelo do Aprendiz:
  - M1;
  - M2;
  - M3.
- Tipo de Modelo:
  - Deterministico;
  - Nao-Deterministico.
- Política Pedagógica:
  - P1;
  - P2.
- Número de Passos:
  - 300;
  - 500;
  - 1000.
- Métrica:
  - Q (qualidade da ação, tabela $Q(s, a)$);
  - Reforco (reforço);
  - ReforcoMedio (reforço médio);
  - TotVisitas (total de visita a cada estado);
  - TransEst (transição entre os estados).

5 RESULTADOS

Neste capítulo será apresentada a comparação dos resultados obtidos por cada metaheurística aplicada na fase de exploração do algoritmo *Q-Learning* no protótipo de Tutor. Em seguida, encontra-se a discussão da hipótese e análise dos resultados estatísticos. Finalmente, a discussão dos resultados obtidos é apresentada.

5.1 Comparação das Metaheurísticas

Foram realizadas simulações para os modelos de aprendiz **M1** – **Ruim**, **M2** – **Bom** e **M3** – **Excelente**. Para cada um desses modelos de aprendiz foram realizadas simulações de **300**, **500** e **1000** passos. Os resultados apresentados foram obtidos através da média de 20 execuções de cada simulação. Os gráficos e tabelas apresentados nas seções seguintes foram produzidos através das informações apresentadas no Apêndice B.

5.1.1 Modelo de Aprendiz M1 – Ruim

A Figura 20 apresenta a comparação de desempenho do valor de utilidade $Q(s, a)$ de um par (estado(s), ação(a)) das heurísticas Aleatória, Tabu e GRASP aplicadas como estratégia de exploração numa simulação de **300** passos aplicada no modelo de aprendiz **M1** – **Ruim**. A heurística Tabu obteve os melhores resultados de $Q(s, a)$, seguida, respectivamente, pelas heurísticas GRASP e Aleatória.

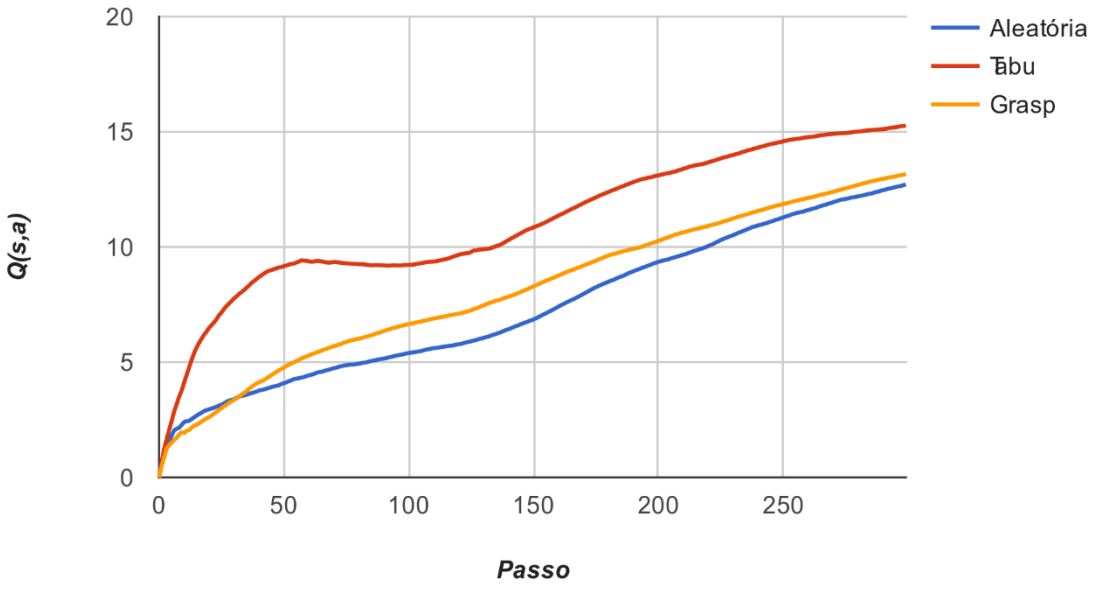

Figura 20 – Comparação das estratégias de exploração Aleatória, Tabu e GRASP – Modelo M1 – 300 passos.

A Tabela 2 apresenta o percentual de visitas nos estados $E0, E1, E2, E3, E4$ das simulações utilizando as heurísticas Aleatória, Tabu e GRASP aplicadas como estratégia de exploração realizados no modelo de aprendiz **M1** – **Ruim** numa simulação de **300** passos.

Tabela 2 – Percentual de Visitas nos Estados por metaheurística – Modelo M1 – 300 passos.

Visitas nos Estados			
Estado	**Aleatória**	**Tabu**	**GRASP**
E0	84,40%	83,62%	86,05%
E1	8,80%	7,02%	6,88%
E2	3,80%	5,07%	4,02%
E3	0,15%	0,25%	0,17%
E4	2,85%	4,05%	2,88%

A Figura 21 apresenta a comparação de desempenho do valor de utilidade $Q(s, a)$ de um par (estado(s), ação(a)) das heurísticas Aleatória, Tabu e GRASP aplicadas como estratégia de exploração numa simulação de **500** passos aplicada no modelo de aprendiz **M1** – **Ruim**. A heurística Tabu obteve os melhores resultados de $Q(s, a)$, seguida, respectivamente, pelas heurísticas GRASP e Aleatória.

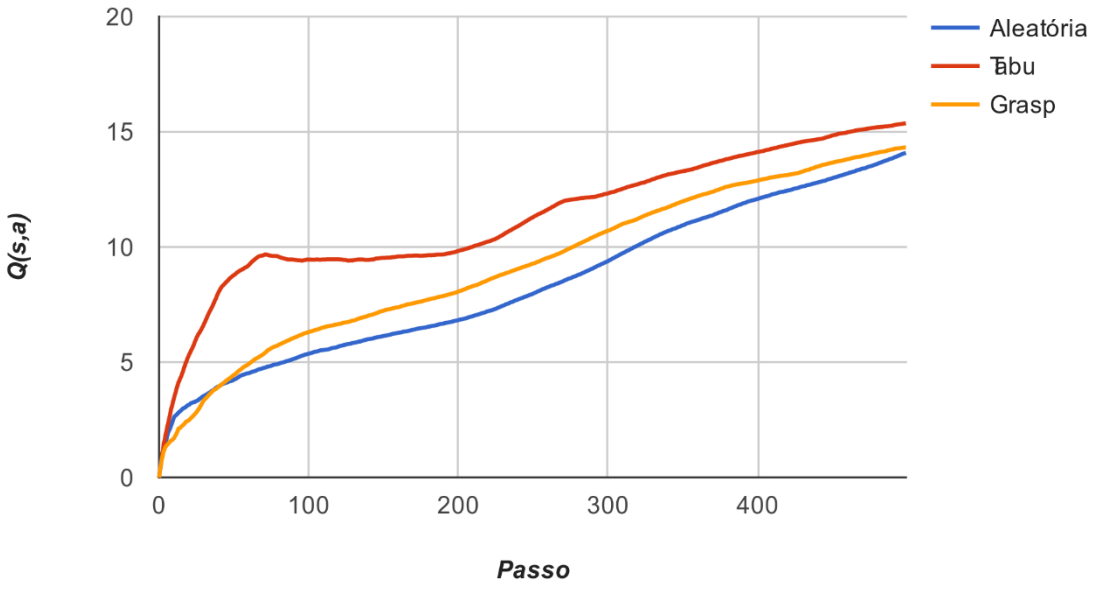

Figura 21 – Comparação das estratégias de exploração Aleatória, Tabu e GRASP – Modelo M1 – 500 passos.

A Tabela 3 apresenta o percentual de visitas nos estados $E0, E1, E2, E3, E4$ das simulações utilizando as heurísticas Aleatória, Tabu e GRASP aplicadas como estratégia de exploração realizados no modelo de aprendiz **M1** – **Ruim** numa simulação de **500** passos.

Tabela 3 – Percentual de Visitas nos Estados por metaheurística – Modelo M1 – 500 passos.

Visitas nos Estados			
Estado	**Aleatória**	**Tabu**	**GRASP**
E0	84,53%	85,72%	86,21%
E1	8,53%	6,51%	7,38%
E2	4,18%	4,45%	3,72%
E3	0,18%	0,20%	0,16%
E4	2,58%	3,12%	2,53%

A Figura 22 apresenta a comparação de desempenho do valor de utilidade $Q(s,a)$ de um par (estado(s), ação(a)) das heurísticas Aleatória, Tabu e GRASP aplicadas como estratégia de exploração numa simulação de 100 passos aplicada no modelo de aprendiz **M1** – **Ruim**. A heurística Tabu obteve os melhores resultados de $Q(s,a)$ seguida, respectivamente, pelas heurísticas GRASP e Aleatória.

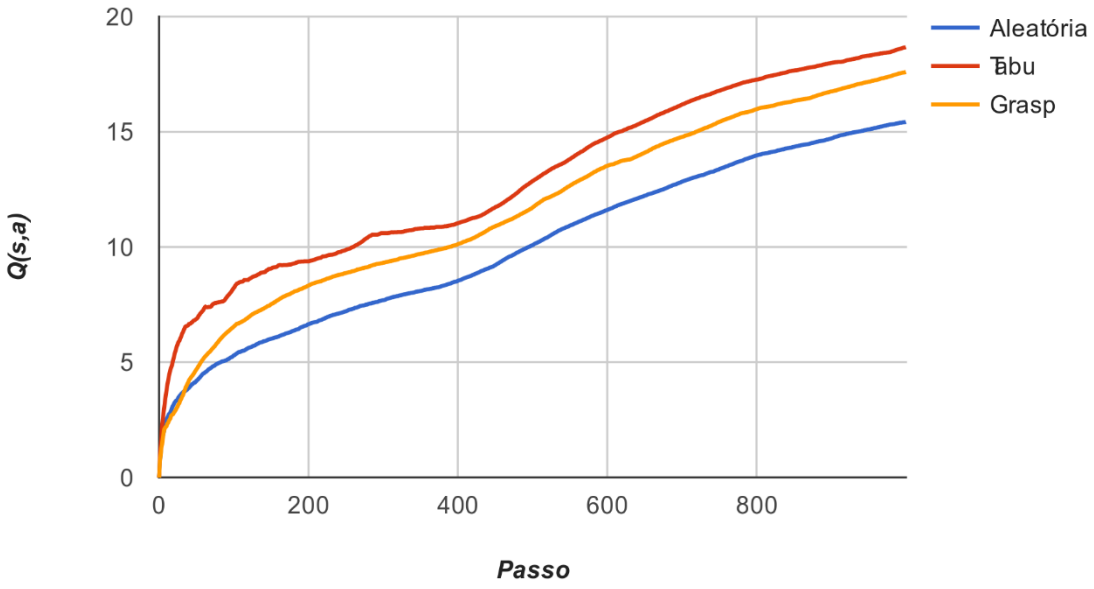

Figura 22 – Comparação das estratégias de exploração Aleatória, Tabu e GRASP – Modelo M1 – 1000 passos.

A Tabela 4 apresenta o percentual de visitas nos estados $E0, E1, E2, E3, E4$ das simulações utilizando as heurísticas Aleatória, Tabu e GRASP aplicadas como estratégia de exploração realizados no modelo de aprendiz **M1** – **Ruim** numa simulação de **1000** passos.

Tabela 4 – Percentual de Visitas nos Estados por metaheurística – Modelo M1 – 1000 passos.

Visitas nos Estados			
Estado	**Aleatória**	**Tabu**	**GRASP**
E0	85,74%	82,20%	83,68%
E1	8,55%	7,60%	8,57%
E2	3,26%	5,84%	4,19%
E3	0,17%	0,18%	0,12%
E4	2,29%	4,19%	3,45%

5.1.2 Modelo de Aprendiz M2 – Bom

A Figura 23 apresenta a comparação de desempenho do valor de utilidade $Q(s, a)$ de um par (estado(s), ação(a)) das heurísticas Aleatória, Tabu e GRASP aplicadas como estratégia de exploração numa simulação de **300** passos aplicada no modelo de aprendiz **M2** –

Bom. A heurística Tabu obteve os melhores resultados de $Q(s,a)$, seguida, respectivamente, pelas heurísticas GRASP e Aleatória.

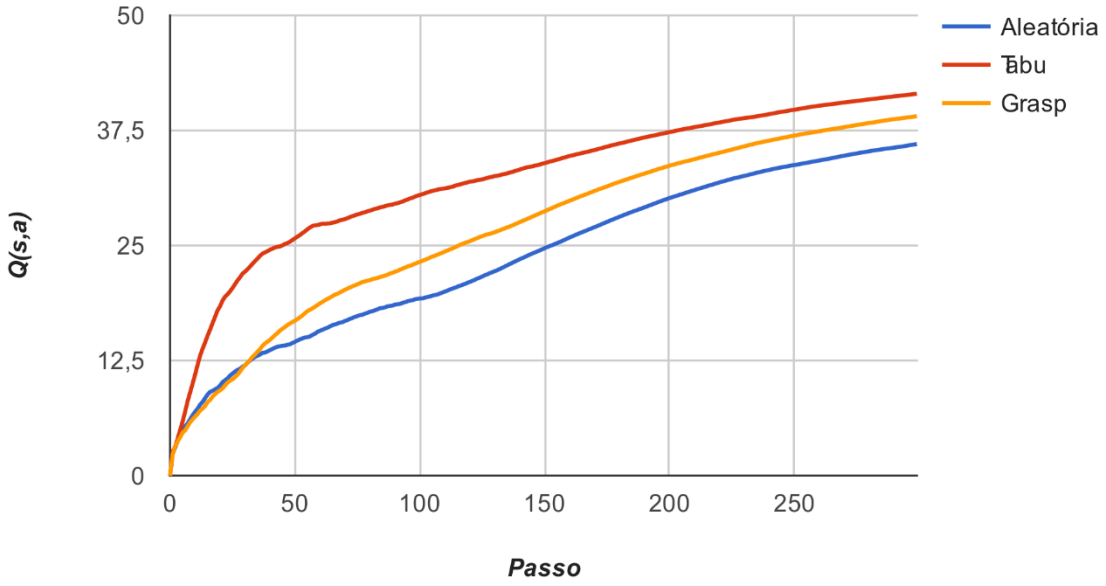

Figura 23 – Comparação das estratégias de exploração Aleatória, Tabu e GRASP – Modelo M2 – 300 passos.

A Tabela 5 apresenta o percentual de visitas nos estados $E0, E1, E2, E3, E4$ das simulações utilizando as heurísticas Aleatória, Tabu e GRASP aplicadas como estratégia de exploração realizados no modelo de aprendiz **M2** – **Bom** numa simulação de **300** passos.

Tabela 5 – Percentual de Visitas nos Estados por metaheurística – Modelo M2 – 300 passos.

Visitas nos Estados			
Estado	**Aleatória**	**Tabu**	**GRASP**
E0	0,98%	1,02%	0,92%
E1	6,10%	5,02%	5,57%
E2	79,57%	74,82%	80,63%
E3	13,22%	18,90%	12,73%
E4	0,13%	0,25%	0,15%

A Figura 24 apresenta a comparação de desempenho do valor de utilidade $Q(s,a)$ de um par (estado(s), ação(a)) das heurísticas Aleatória, Tabu e GRASP aplicadas como estratégia de exploração numa simulação de **500** passos aplicada no modelo de aprendiz **M2** –

Bom. A heurística Tabu obteve os melhores resultados de $Q(s,a)$, seguida, respectivamente, pelas heurísticas GRASP e Aleatória.

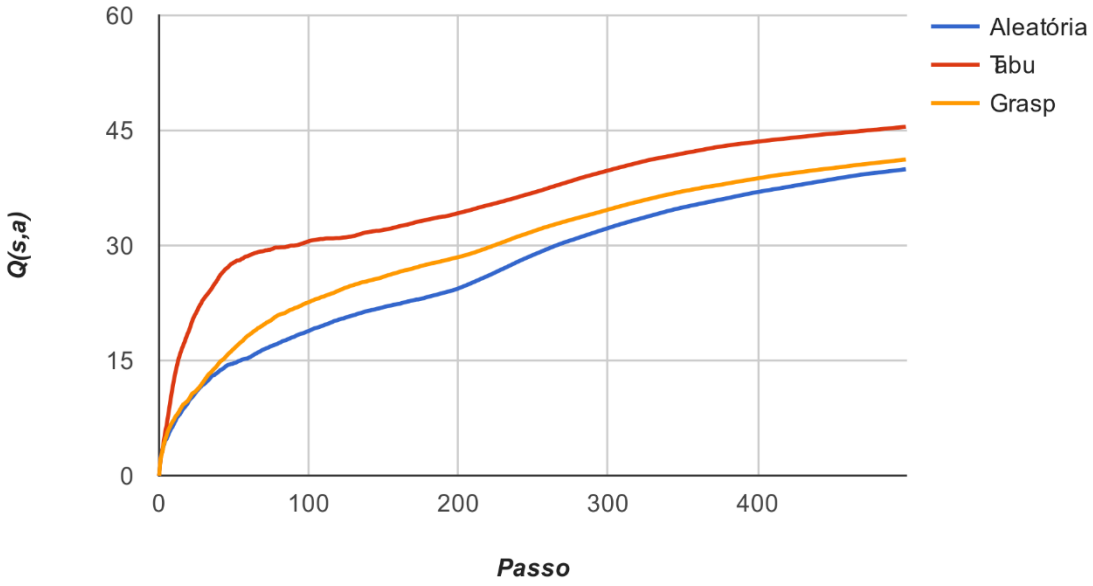

Figura 24 – Comparação das estratégias de exploração Aleatória, Tabu e GRASP – Modelo M2 – 500 passos.

A Tabela 6 apresenta o percentual de visitas nos estados $E0, E1, E2, E3, E4$ das simulações utilizando as heurísticas Aleatória, Tabu e GRASP aplicadas como estratégia de exploração realizados no modelo de aprendiz **M2** – **Bom** numa simulação de **500** passos.

Tabela 6 – Percentual de Visitas nos Estados por metaheurística – Modelo M2 – 500 passos.

Visitas nos Estados			
Estado	**Aleatória**	**Tabu**	**GRASP**
E0	0,72%	0,62%	0,81%
E1	4,63%	4,64%	5,43%
E2	79,58%	65,79%	80,89%
E3	14,85%	28,71%	12,66%
E4	0,22%	0,24%	0,21%

A Figura 25 apresenta a comparação de desempenho do valor de utilidade $Q(s,a)$ de um par (estado(s), ação(a)) das heurísticas Aleatória, Tabu e GRASP aplicadas como estratégia de exploração numa simulação de **1000** passos aplicada no modelo de aprendiz **M2**

– **Bom**. A heurística Tabu obteve os melhores resultados de $Q(s, a)$, seguida, respectivamente, pelas heurísticas GRASP e Aleatória.

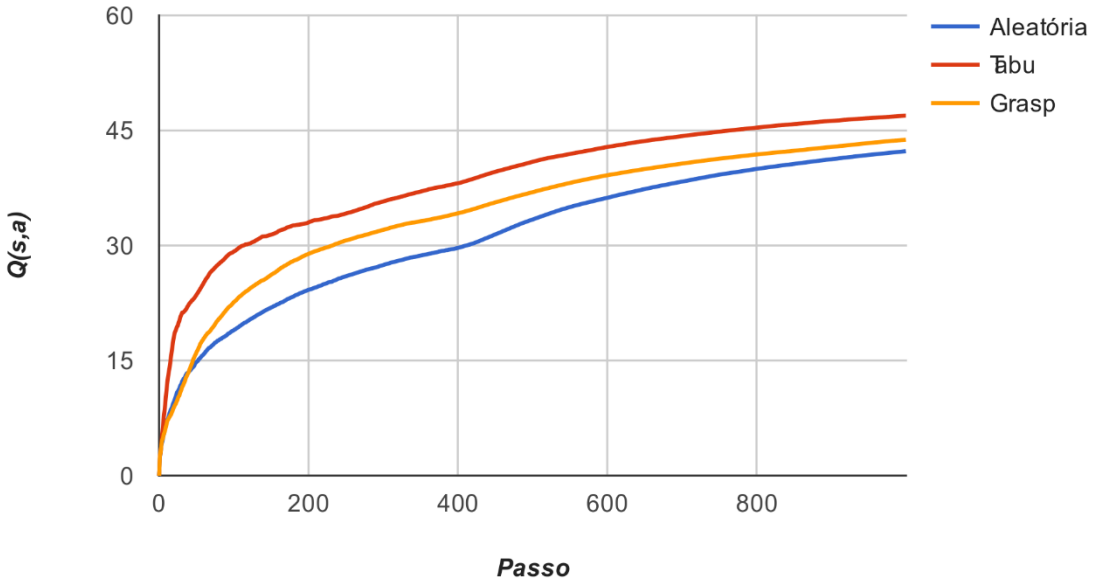

Figura 25 – Comparação das estratégias de exploração Aleatória, Tabu e GRASP – Modelo M2 – 1000 passos.

A Tabela 7 apresenta o percentual de visitas nos estados $E0, E1, E2, E3, E4$ das simulações utilizando as heurísticas Aleatória, Tabu e GRASP aplicadas como estratégia de exploração realizados no modelo de aprendiz **M2** – **Bom** numa simulação de **1000** passos.

Tabela 7 – Percentual de Visitas nos Estados por metaheurística – Modelo M2 – 1000 passos.

Visitas nos Estados			
Estado	**Aleatória**	**Tabu**	**GRASP**
E0	0,53%	0,52%	0,57%
E1	5,42%	2,66%	6,41%
E2	82,12%	83,52%	83,69%
E3	11,76%	13,14%	9,12%
E4	0,18%	0,17%	0,22%

5.1.3 Modelo de Aprendiz M3 – Excelente

A Figura 26 apresenta a comparação de desempenho do valor de utilidade $Q(s,a)$ de um par (estado(s), ação(a)) das heurísticas Aleatória, Tabu e GRASP aplicadas como estratégia de exploração numa simulação de **300** passos aplicada no modelo de aprendiz **M3** – **Excelente**. A heurística Tabu obteve os melhores resultados de $Q(s,a)$, seguida, respectivamente, pelas heurísticas GRASP e Aleatória.

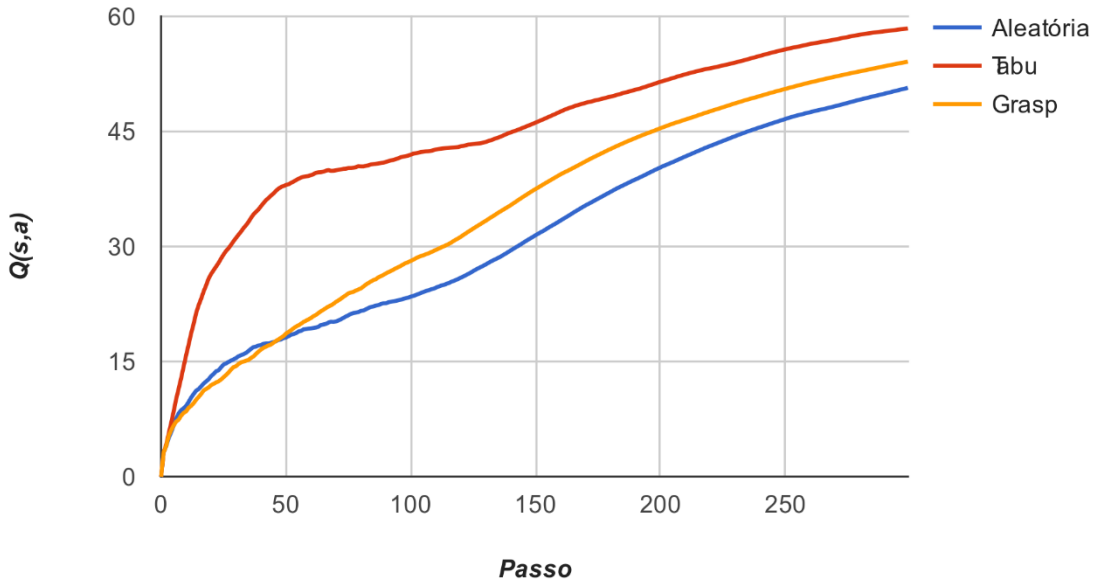

Figura 26 – Comparação das estratégias de exploração Aleatória, Tabu e GRASP – Modelo M3 – 300 passos.

A Tabela 8 apresenta o percentual de visitas nos estados $E0, E1, E2, E3, E4$ das simulações utilizando as heurísticas Aleatória, Tabu e GRASP aplicadas como estratégia de exploração realizados no modelo de aprendiz **M3** – **Excelente** numa simulação de **1000** passos.

Tabela 8 – Percentual de Visitas nos Estados por metaheurística – Modelo M3 – 300 passos.

Visitas nos Estados			
Estado	**Aleatória**	**Tabu**	**GRASP**
E0	0,93%	1,07%	0,97%
E1	1,40%	1,63%	0,87%
E2	16,47%	17,33%	7,28%
E3	35,25%	32,98%	46,67%
E4	45,95%	46,98%	44,22%

A Figura 27 apresenta a comparação de desempenho do valor de utilidade $Q(s,a)$ de um par (estado(s), ação(a)) das heurísticas Aleatória, Tabu e GRASP aplicadas como estratégia de exploração numa simulação de **500** passos aplicada no modelo de aprendiz **M3** – **Excelente**. A heurística Tabu obteve os melhores resultados de $Q(s,a)$, seguida, respectivamente, pelas heurísticas GRASP e Aleatória.

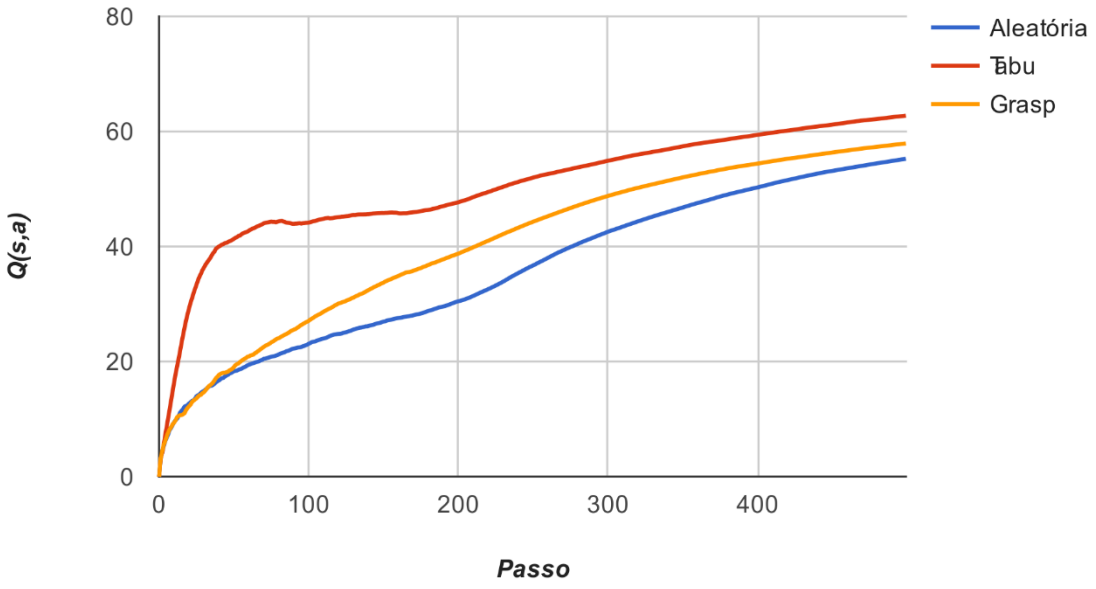

Figura 27 – Comparação das estratégias de exploração Aleatória, Tabu e GRASP – Modelo M3 – 500 passos.

A Tabela 9 apresenta o percentual de visitas nos estados $E0, E1, E2, E3, E4$ das simulações utilizando as heurísticas Aleatória, Tabu e GRASP aplicadas como estratégia de exploração realizados no modelo de aprendiz **M3** – **Excelente** numa simulação de **500** passos.

Tabela 9 – Percentual de Visitas nos Estados por metaheurística – Modelo M3 – 500 passos.

Visitas nos Estados			
Estado	**Aleatória**	**Tabu**	**GRASP**
E0	0,62%	0,57%	0,86%
E1	1,41%	1,62%	1,01%
E2	18,78%	16,65%	7,67%
E3	30,24%	32,33%	47,62%
E4	48,95%	48,83%	42,84%

A Figura 28 apresenta a comparação de desempenho do valor de utilidade $Q(s,a)$ de um par (estado(s), ação(a)) das heurísticas Aleatória, Tabu e GRASP aplicadas como estratégia de exploração numa simulação de **1000** passos aplicada no modelo de aprendiz **M3** – **Excelente**. A heurística Tabu obteve os melhores resultados de $Q(s,a)$, seguida, respectivamente, pelas heurísticas GRASP e Aleatória.

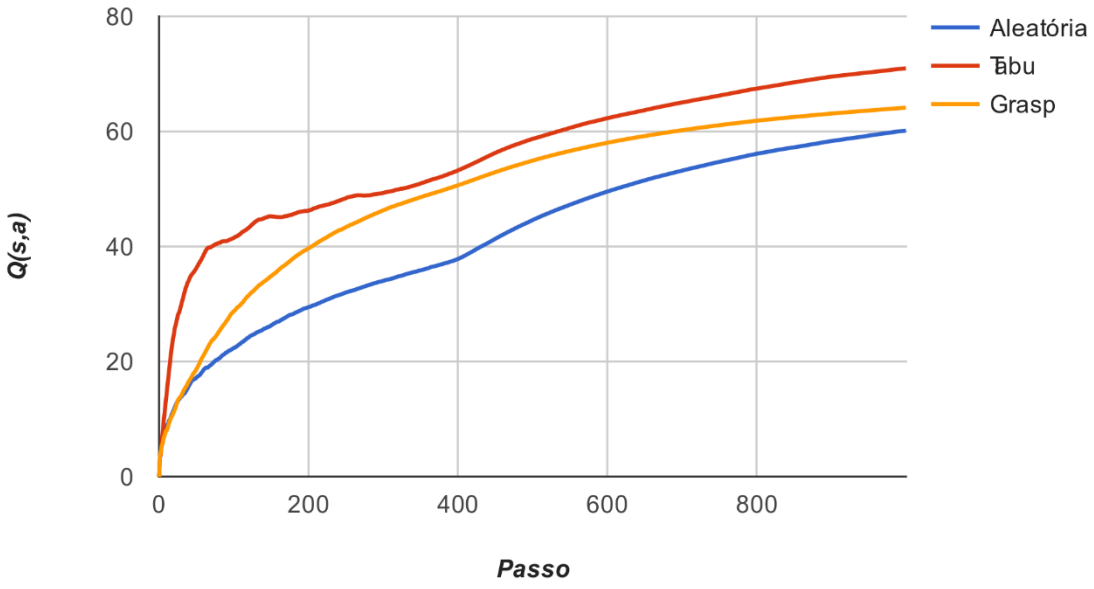

Figura 28 – Comparação das estratégias de exploração Aleatória, Tabu e GRASP – Modelo M3 – 1000 passos.

A Tabela 10 apresenta o percentual de visitas nos estados $E0, E1, E2, E3, E4$ das simulações utilizando as heurísticas Aleatória, Tabu e GRASP aplicadas como estratégia de exploração realizados no modelo de aprendiz **M3** – **Excelente** numa simulação de **1000** passos.

Tabela 10 – Percentual de Visitas nos Estados por metaheurística – Modelo M3 – 1000 passos.

Visitas nos Estados			
Estado	**Aleatória**	**Tabu**	**GRASP**
E0	0,45%	0,42%	0,55%
E1	1,55%	1,12%	0,86%
E2	15,99%	8,07%	5,01%
E3	31,82%	27,80%	48,10%
E4	50,21%	62,60%	45,50%

5.2 Discussão da Hipótese

A hipótese nula deste trabalho consiste na afirmação de que a inclusão de metaheurísticas em sistemas tutores inteligentes, com a característica de modelagem autônoma do aprendiz, não conseguem melhorar a convergência do algoritmo *Q-Learning*, utilizado para aprender quais as melhores ações (estratégias pedagógicas) a serem adotadas dada um estado cognitivo do aprendiz.

Formalmente, pode-se apresentar essa hipótese nula através da seguinte hipótese:

$$H_0 = K_{Q-Learning\ sem\ metaheurísticas} = K_{Q-Learning\ com\ metaheurísticas}$$

Onde:

H_0= hipótese nula

$K_{Q-Learning\ sem\ metaheurísticas}$ = Algoritmo *Q-Learning* sem introdução de metaheurísticas (fase de exploração Aleatória e fase de explotação Gulosa).

$K_{Q-Learning\ com\ metaheurísticas}$ = Algoritmo *Q-Learning* com introdução de metaheurísticas (fase de exploração com metaheurística e fase de explotação Gulosa).

Se a hipótese nula for considerada falsa, alguma outra afirmativa deve ser verdadeira. Este trabalho propõe a hipótese alternativa H_1, na qual uma metaheurística é introduzida na fase de exploração do algoritmo *Q-Learning*, levando a melhoria na convergência do algoritmo, conseguindo maiores resultados de qualidade das ações escolhidas e maior percentual de visitas a estados cognitivos superiores.

A hipótese alternativa está formalmente representada através da seguinte equação:

$$H_1 = K_{Q-Learning\ com\ metaheurísticas} > K_{Q-Learning\ sem\ metaheurísticas}$$

A metodologia de teste de hipótese neste trabalho considerou os resultados das simulações de um protótipo de sistema tutor inteligente interagindo com um simulador de aprendiz. As simulações foram realizadas com os modelos de aprendiz **M1** – **Ruim**, **M2** – **Bom** e **M3** – **Excelente**. Para cada um desses modelos de aprendiz foram realizadas simulações de **300**, **500** e **1000** passos. Para cada modelo de aprendiz e quantidade de passos foram realizadas simulações comparando o resultado obtido de qualidade do valor de utilidade $Q(s,a)$ de um par (estado(s), ação(a)) e percentual de visitas nos estados $E0, E1, E2, E3, E4$ das heurísticas Aleatória, Tabu e GRASP.

Assim sendo, foi utilizado o teste ANOVA de Friedman (CALLEGARI-JACQUES, 2007), que considera que as diversas amostras são, estatisticamente, idênticas, na sua distribuição (hipótese de nulidade, ou de H_0). A hipótese alternativa (H_1) aponta como elas são significativamente diferentes, na sua distribuição e o teste de concordância de Kendall (CALLEGARI-JACQUES, 2007) normaliza o teste estatístico de Friedman, com a finalidade de gerar uma avaliação de concordância, ou não, com ranques estabelecidos.

5.3 Análise dos Resultados Estatísticos

Foram geradas tabelas para cada um dos modelos de aprendiz, para análise da relevância estatística da diferença de desempenho da qualidade das ações escolhidas pelo simulador de tutor realizando a exploração aleatória e com a introdução das metaheurísticas Busca Tabu e GRASP.

As tabelas comparam o desempenho da métrica valor de utilidade $Q(s,a)$ de um par (estado(s), ação(a)) das heurísticas nas simulações de **300**, **500** e **1000** passos para cada um dos modelos de aprendiz. As tabelas são compostas de informações calculadas pelo teste estatístico ANOVA de Friedman e Coeficiente de Concordância de Kendall, os valores de Ordem Médio, Soma de Ordem, Média e Desvio Padrão são apresentados para as simulações. A Tabela 11, nas simulações de **300, 500** e **100** passos das heurísticas de exploração Aleatória, Tabu e GRASP. É possível observar que, segundo os testes, a heurística Tabu nas simulações de, respectivamente **300, 500** e **1000** passos, obteve a maior classificação (2,99; 2,99 e 3), maior valor de soma de ordens (898,00; 1496,50 e 2999,00), maior média (11,05; 11,26 e12,98). O coeficiente de concordância de Kendall obtido para os testes foi igual ou acima de 0,9 (0,90; 0,92 e 0,97).

Tabela 12 e Tabela 13 apresentam essas informações, respectivamente para os modelos de aprendiz **M1** – **Ruim**, **M2** – **Bom** e **M3 - Excelente**:

Tabela 11 – Comparação estatística das amostras – Modelo de Aprendiz Ruim – M1 - 300, 500 e 1000 passos

Modelos - Passos	**M1 - 300**	**M1 - 500**	**M1 - 1000**	**M1 - 300**	**M1 - 500**	**M1 - 1000**
	ANOVA de Friedman					
α=0,05				**α(qui-quadrado) p-valor (bilateral)<0,0001**		
GL=2	-	-	-	538,30	917,05	1935,60
Exploração	**Ordem médio**			**Soma de Ordens**		
Aleatória	1,11	1,09	1,03	333,50	543,00	1033,50
Tabu	2,99	2,99	3,00	898,00	1496,50	2999,00
GRASP	1,90	1,92	1,97	568,50	960,50	1967,50
	Coeficiente de Concordância de Kendall					
α=0,05	**Coef. de Concordância de Kendall**			**Ordem Médio**		
GL=2	0,90	0,92	0,97	0,90	0,92	0,97
Exploração	**Média**			**Desvio Padrão**		
Aleatória	7,37	8,39	10,05	3,19	3,33	3,59
Tabu	11,05	11,26	12,98	3,11	2,88	3,89
GRASP	8,17	9,24	11,70	3,30	3,48	4,00

A Tabela 11 apresenta as comparações entre os resultados obtidos nas simulações realizadas com o modelo de aprendiz **M1** – **Ruim** nas simulações de **300**, **500** e 100 passos das heurísticas de exploração Aleatória, Tabu e GRASP. É possível observar que, segundo os testes, a heurística Tabu nas simulações de, respectivamente **300**, **500** e **1000** passos, obteve a maior classificação (2,99; 2,99 e 3), maior valor de soma de ordens (898,00; 1496,50 e 2999,00), maior média (11,05; 11,26 e12,98). O coeficiente de concordância de Kendall obtido para os testes foi igual ou acima de 0,9 (0,90; 0,92 e 0,97).

Tabela 12 - Comparação estatística das amostras – Modelo de Aprendiz Bom - M2 - 300, 500 e 1000 passos

Modelos - Passos	**M2 - 300**	**M2 - 500**	**M2 - 1000**	**M2 - 300**	**M2 - 500**	**M2 - 1000**
	ANOVA de Friedman					
α=0,05				**α(qui-quadrado) p-valor (bilateral)<0,0001**		
GL=2	-	-	-	544,70	986,05	1930,32
Exploração	**Ordem médio**			**Soma de Ordens**		
Aleatória	1,10	1,01	1,04	329,50	505,00	1036,00
Tabu	2,99	2,99	3,00	898,00	1497,00	2998,00
GRASP	1,91	2,00	1,97	572,50	998,00	1966,00
	Coeficiente de Concordância de Kendall					
α=0,05	**Coef. de Concordância de Kendall**			**Ordem Médio**		
GL=2	0,91	0,99	0,97	0,91	0,99	0,97
Exploração	**Média**			**Desvio Padrão**		
Aleatória	23,90	27,43	31,53	8,81	9,35	8,81
Tabu	32,19	35,92	38,68	8,22	8,09	7,73
GRASP	26,88	29,96	34,52	9,68	9,21	8,65

A Tabela 12 apresenta as comparações entre os resultados obtidos nas simulações realizadas com o modelo de aprendiz **M2** – **Bom** nas simulações de **300**, **500** e 100 passos das heurísticas de exploração Aleatória, Tabu e GRASP. É possível observar que, segundo os testes, a heurística Tabu nas simulações de, respectivamente **300**, **500** e **1000** passos, obteve a maior classificação (2,99; 2,99 e 3), maior valor de soma de ordens (898,00; 1497,00 e 2998,00), maior média (32,19; 35,92 e 38,68). O coeficiente de concordância de Kendall obtido para os testes foi igual acima de 0,91 (0,91; 0,99 e 0,97).

Tabela 13 - Comparação estatística das amostras – Modelo de Aprendiz Bom - M2 - 300, 500 e 1000 passos

Modelos - Passos	**M3 - 300**	**M3 - 500**	**M3 - 1000**	**M3 - 300**	**M3 - 500**	**M3 - 1000**
	ANOVA de Friedman					
α=0,05				**α(qui-quadrado) p-valor (bilateral)<0,0001**		
GL=2	-	-	-	526,74	947,71	1946,98
Exploração	**Ordem médio**			**Soma de Ordens**		
Aleatória	1,14	1,05	1,03	342,00	524,00	1025,00
Tabu	2,99	2,99	3,00	898,00	1495,00	2996,00
GRASP	1,87	1,96	1,98	560,00	981,00	1979,00
	Coeficiente de Concordância de Kendall					
α=0,05	**Coef. de Concordância de Kendall**			**Ordem Médio**		
GL=2	0,88	0,95	0,97	0,88	0,95	0,97
Exploração	**Média**			**Desvio Padrão**		
Aleatória	31,53	35,95	42,10	12,75	13,48	13,85
Tabu	44,83	50,37	56,12	11,28	10,36	12,07
GRASP	35,06	40,72	50,03	14,24	14,21	14,03

A Tabela 13 apresenta as comparações entre os resultados obtidos nas simulações realizadas com o modelo de aprendiz **M3** – **Excelente** nas simulações de **300**, **500** e **1000** passos das heurísticas de exploração Aleatória, Tabu e GRASP. É possível observar que, segundo os testes, a heurística Tabu nas simulações de, respectivamente **300**, **500** e **1000** passos, obteve a maior classificação (2,99; 2,99 e 3), maior valor de soma de ordens (898,00; 1495,00 e 2996,00), maior média (44,83; 50,37 e 56,12). O coeficiente de concordância de Kendall obtido para os testes foi igual acima de 0,88 (0,88; 0,95 e 0,97).

Nos testes estatísticos, apresentados na Tabela 11, nas simulações de **300, 500** e 100 passos das heurísticas de exploração Aleatória, Tabu e GRASP. É possível observar que, segundo os testes, a heurística Tabu nas simulações de, respectivamente **300, 500** e **1000** passos, obteve a maior classificação (2,99; 2,99 e 3), maior valor de soma de ordens (898,00; 1496,50 e 2999,00), maior média (11,05; 11,26 e12,98). O coeficiente de concordância de Kendall obtido para os testes foi igual ou acima de 0,9 (0,90; 0,92 e 0,97).

Tabela 12 e Tabela 13, através da classificação estatística, a heurística Busca Tabu apresentou os melhores resultados nas simulações para todos os modelos seguida pela heurística GRASP, a exploração Aleatória foi classificada em terceiro lugar. Portanto observou-se a rejeição da hipótese nula (H_0) e a aceitação da hipótese alternativa (H_1) deste trabalho com um grau de significância de p-valor (bilateral) < 0,001.

5.4 Discussão dos Resultados

Nas simulações realizadas com o protótipo de sistema tutor inteligente o algoritmo *Q-Learning* é dividido em duas fases: fase de exploração e fase de explotação. Para as simulações realizadas, o algoritmo foi configurado com a exploração de 20% do número de passos, ou seja, para as simulações de **300**, **500** e **1000** passos, respectivamente, são realizados 60, 100 e 200 passos exploratórios. Ao final da execução desses passos a fase exploratória do algoritmo é desativada e o algoritmo realizará apenas a explotação das melhores ações identificadas para cada estado, ou seja, escolherá a ação com o maior valor de utilidade. Durante cada simulação, a cada passo o algoritmo alterna, aleatoriamente, com a probabilidade de escolha de 50% entre as fases de exploração e explotação. Portanto, em média, durante os 40% primeiros passos de cada simulação a fase exploratória do algoritmo ainda está ativada.

As heurísticas Busca Tabu e GRASP foram aplicadas durante a fase exploratória do algoritmo *Q-Learning*. Conforme os testes estatísticos realizados na seção 4.3, Análise dos Resultados Estatísticos, é possível afirmar que, estatisticamente, existe relevância entre a diferença dos resultados obtidos nas simulações realizados com a introdução das metaheurísticas Busca Tabu, GRASP em relação à heurística aleatória. Ainda é possível afirmar que, estatisticamente, é possível classificar o desempenho da qualidade das ações escolhidas entre as heurísticas pela seguinte ordem, do melhor para o pior desempenho, Busca Tabu, GRASP e, finalmente, Aleatória.

Uma das chaves para o melhor desempenho da heurística Busca Tabu em relação às demais heurísticas é o do critério de aspiração adotado. O conceito de “melhor resultado até agora”, proposto em Zhang e Liu (2008), permite ao algoritmo que, mesmo durante a fase de exploração, quando valores muito promissores são encontrados durante a fase inicial da simulação os mesmos possam ser adotados. Na metaheurística GRASP, durante a fase de construção uma solução é gerada e, na fase seguinte, a lista restrita de candidatos é produzida selecionando potenciais soluções promissoras na vizinhança. A solução a ser adotada é escolhida aleatoriamente entre as soluções identificadas nessa lista. A heurística Aleatória, por

sua vez, apenas escolhe aleatoriamente uma das possíveis soluções, sem nenhum mecanismo para priorizar potenciais melhores soluções na fase de exploração, o que justifica o seu pior desempenho em relação às demais heurísticas.

Os gráficos disponíveis na Figura 29, Figura 30 e Figura 31 foram produzidos através da sobreposição dos resultados das heurísticas Aleatória, Busca Tabu e GRASP nas simulações de, respectivamente **300**, **500** e **1000** passos. Se analisarmos os resultados das heurísticas a cada grupo de modelo de aprendiz (**M1** – **Ruim**, **M2** – **Bom**, **M3 - Excelente**), é possível notar o desempenho da metaheurística Busca Tabu se destacar das demais heurísticas no mesmo grupo enquanto a fase de exploração do algoritmo *Q-Learning* está ativada.

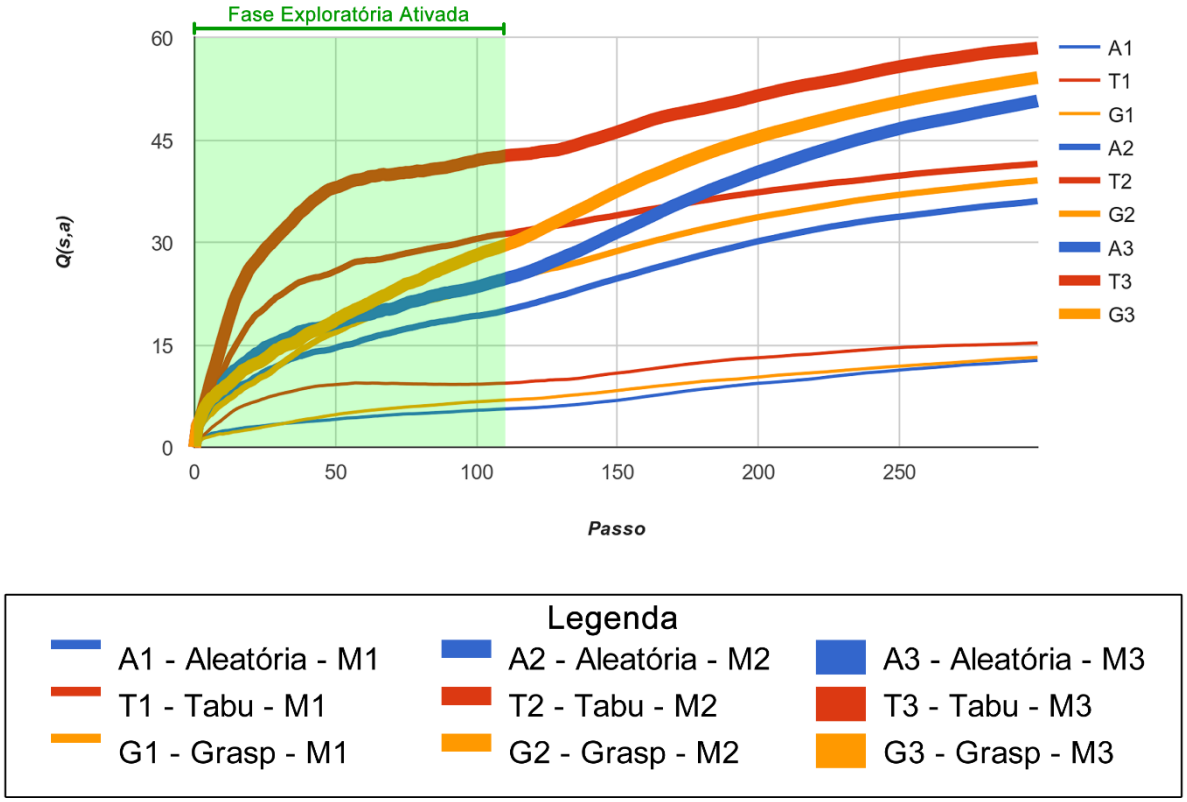

Figura 29 - Comparação das estratégias de exploração Aleatória, Tabu e GRASP – Modelos M1, M2 e M3 – 300 passos.

A Figura 29 apresenta o gráfico comparativo de desempenho da métrica $Q(s,a)$ para as simulações de **300** passos dos modelos **M1** – **Ruim**, **M2** – **Bom** e **M3** – **Excelente**. Em cada um dos modelos, os resultados obtidos pela metaheurística Busca Tabu obteve melhores resultados em relação às demais heurísticas, a heurística GRASP apresenta o segundo melhor desempenho seguida, finalmente, da heurística Aleatória. Conforme pode-se observar, a inclinação da curva de aprendizado se acentua enquanto a fase exploratória do algoritmo está ativada.

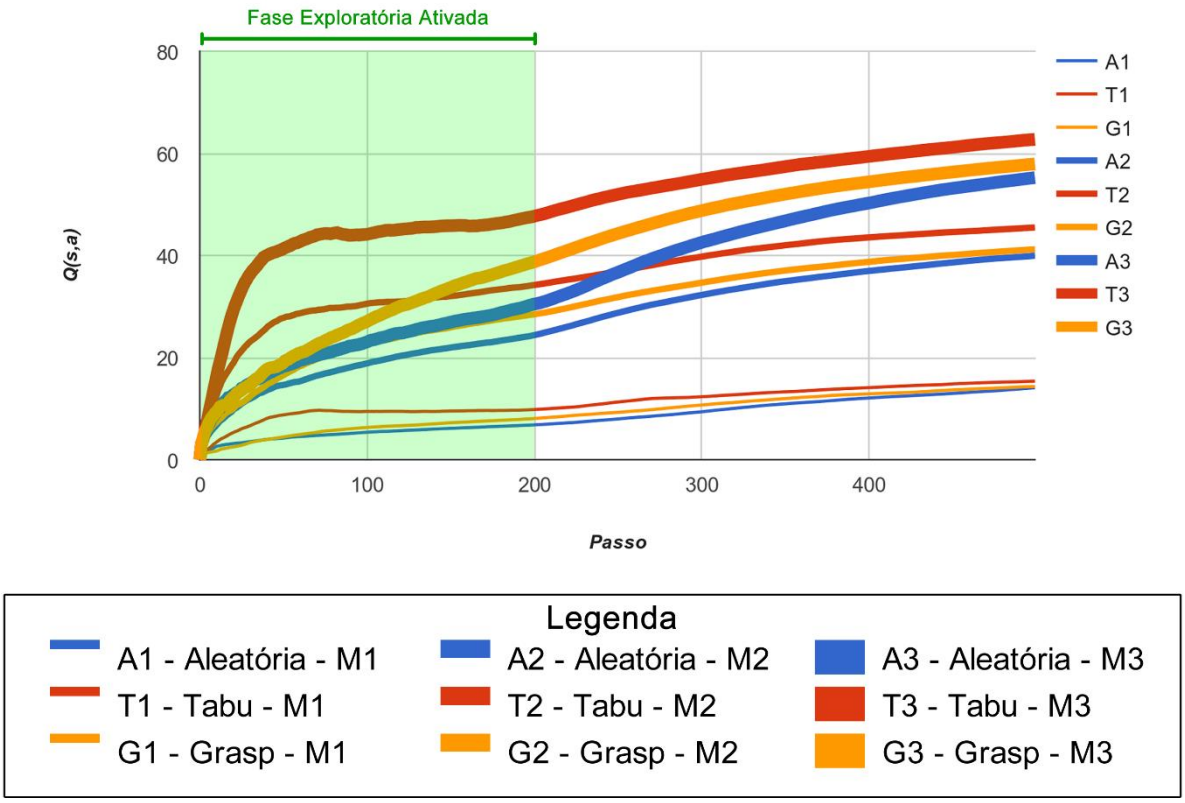

Figura 30 - Comparação das estratégias de exploração Aleatória, Tabu e GRASP – Modelos M1, M2 e M3 – 500 passos.

A Figura 30 apresenta o gráfico comparativo de desempenho da métrica $Q(s,a)$ para as simulações de **500** passos dos modelos **M1** – **Ruim**, **M2** – **Bom** e **M3** – **Excelente**. Em cada um dos modelos, os resultados obtidos pela metaheurística Busca Tabu obteve melhores resultados em relação às demais heurísticas, a heurística GRASP apresenta o segundo melhor desempenho seguida, finalmente, da heurística Aleatória. Conforme pode-se observar, a inclinação da curva de aprendizado se acentua enquanto a fase exploratória do algoritmo está ativada.

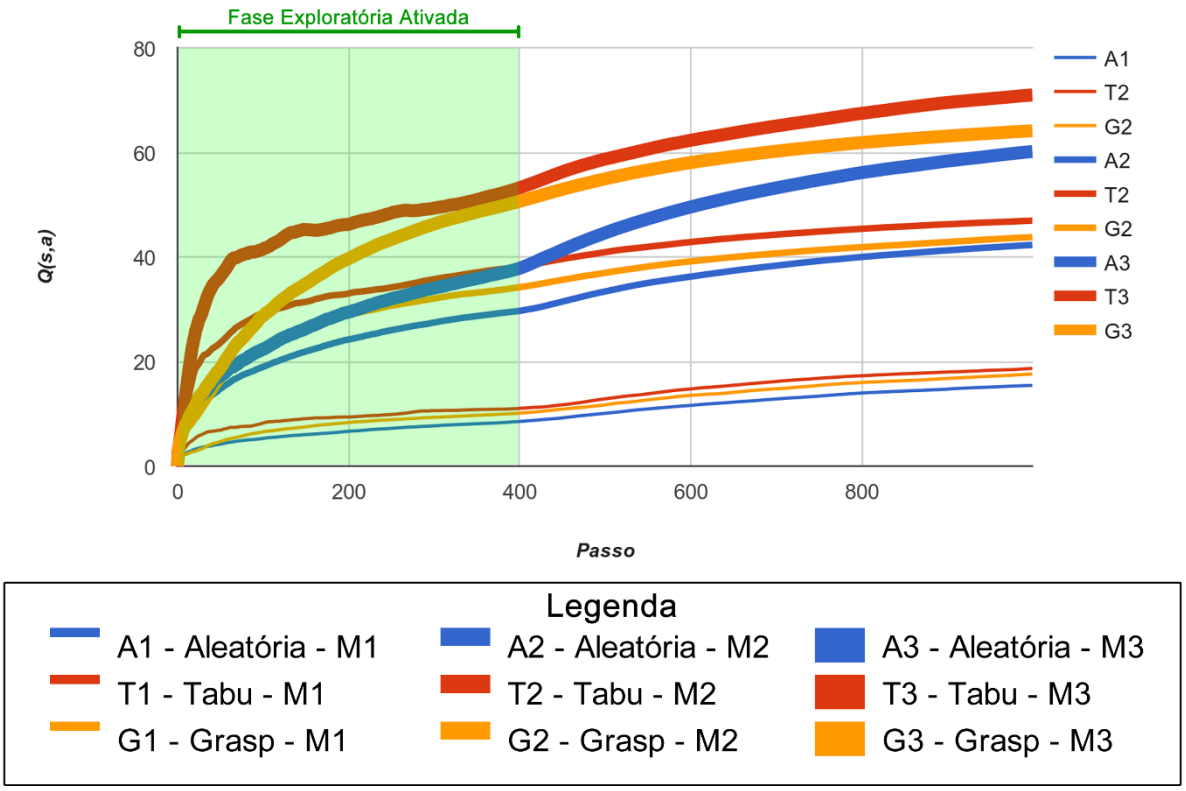

Figura 31 - Comparação das estratégias de exploração Aleatória, Tabu e GRASP – Modelos M1, M2 e M3 – 1000 passos.

A Figura 31 apresenta o gráfico comparativo de desempenho da métrica $Q(s,a)$ para as simulações de **1000** passos dos modelos **M1** – **Ruim**, **M2** – **Bom** e **M3** – **Excelente**. Em cada um dos modelos, os resultados obtidos pela metaheurística Busca Tabu obteve melhores resultados em relação às demais heurísticas, a heurística GRASP apresenta o segundo melhor desempenho seguida, finalmente, da heurística Aleatória. Conforme pode-se observar, a inclinação da curva de aprendizado se acentua enquanto a fase exploratória do algoritmo está ativada.

6 CONCLUSÕES

Neste trabalho foi proposta a introdução das metaheurísticas Busca Tabu e GRASP como política de exploração do algoritmo *Q-Learning* em STIs com característica de modelagem autônoma do aprendiz. Os resultados obtidos indicam que a introdução de metaheurísticas adequadas podem acelerar o aprendizado, e alcançar maiores valores nas métricas de desempenho do algoritmo.

Nas simulações apresentadas neste trabalho, a introdução da Busca Tabu e GRASP como políticas de exploração do algoritmo *Q-Learning* obtiveram aumento, com relevância estatística comprovada, nos valores de qualidade da ação $Q(s,a)$. Essa introdução também teve como consequência um maior número percentual de visitas aos estados cognitivos superiores em relação à política de exploração aleatória. Ou seja, para um mesmo modelo, mais vezes o aprendiz conseguiu sair de um estado cognitivo inferior para um estado cognitivo superior.

Os resultados das simulações realizadas mostraram que a metaheurística Busca Tabu introduzida na fase de exploração do algoritmo *Q-Learning* obteve melhores resultados que a metaheurística GRASP. Por sua vez, a metaheurística GRASP obteve melhores resultados que a heurística Aleatória, utilizada como *baseline* para as simulações realizadas.

Nas simulações realizadas de **300**, **500** e **1000** passos os modelos de aprendiz utilizados (**M1** – **Ruim**, **M2 - Bom** e **M3** – **Excelente**) apresentaram comportamento semelhante. A curva de aprendizagem se mostrou mais acentuada enquanto a exploração está ativada, especialmente nos casos em que as metaheurísticas Tabu e GRASP estão introduzidas, o que demonstra que as metaheurísticas levaram o simulador de STI a escolher ações com maior qualidade quando utiliza metaheurísticas adequadas.

O maior número de visita aos estados acontece dentro das faixas esperadas para os modelos: **M1** – **Ruim**, Estado $E0$; **M2** – **Bom**, Estado $E2$; **M3** – **Excelente**, Estado $E4$. No entanto a heurística Busca Tabu consegue, com maior frequência, levar o aprendiz a estados cognitivos superiores à faixa esperada do modelo em relação à metaheurística GRASP e heurística Aleatória.

Apesar de variar-se o número de passos das simulações (**300**, **500** e **1000**) passos, a inclinação da curva de aprendizagem apresenta uma mesma tendência: maior inclinação enquanto a fase exploratória do algoritmo está ativada e menor inclinação na fase de explotação do algoritmo.

Um mesmo valor de $Q(s,a)$ é atingido pelas simulações com a utilização de metaheurísticas (Busca Tabu e GRASP) com um número inferior de passos em relação às simulações sem metaheurística (Aleatória). Portanto, o algoritmo converge mais rapidamente utilizando metaheurísticas adequadas.

Conforme pode ser visualizado nas Tabela 11, Tabela 12 e Tabela 13 a hipótese foi comprovada nas simulações de **300**, **500** e **1000** passos, dos modelos **M1**, **M2** e **M3** e a diferença entre os resultados apresenta relevância estatística analisada pelos testes ANOVA de Friedman e Coeficiente de Concordância de Kendall.

A aceleração do tempo de convergência do algoritmo *Q-Learning* pode otimizar a utilização do algoritmo alcançando maior valor de qualidade para as ações ou mesmo viabilizar a utilização de sistemas tutores inteligentes em ambientes em que, a princípio, não existem modelos do aprendiz disponíveis (PAIVA; GUELPELI, 2016).

6.1 Contribuições

A partir dos experimentos e das análises realizadas nos resultados, podem-se destacar algumas contribuições importantes para a área de Inteligência Artificial, Aprendizagem por Reforço, Sistemas Tutores Inteligentes, especificamente utilização de metaheurísticas:

- O modelo proposto neste trabalho que permite a inclusão de metaheurísticas tanto na fase de exploração do algoritmo *Q-Learning* quanto na sua fase de explotação.
- Para realizar a comparação dos resultados obtidos nas simulações realizadas no protótipo de sistema tutor inteligente, o programa original foi alterado do paradigma de programação estruturada para programação orientada a objetos. Utilizou-se o Ambiente Desenvolvimento Integrado QtCreator para possibilitar através de interface gráfica modificar as configurações das simulações e exibir os resultados comparativos obtidos por cada metaheurística adotada. Esse simulador permite a realização de simulações tanto com as metaheurísticas descritas neste trabalho como inclusão de outras metaheurísticas.
- Os resultados obtidos comprovaram que é possível melhorar o tempo de convergência do algoritmo *Q-Learning* através da inserção de metaheurísticas em sistemas tutores inteligentes com característica de modelagem autônoma do aprendiz.

6.2 Limitações

Nas simulações realizadas no presente trabalho, não foram realizados experimentos com a política pedagógica **P2**, uma política mais restritiva do que **P1** em relação ao modelo.

As metaheurísticas Busca Tabu e GRASP foram adotadas somente na fase exploratória do Algoritmo *Q-Learning*. Em todas as simulações apresentadas, adotou-se somente a heurística gulosa na fase de explotação do algoritmo.

6.3 Trabalhos Futuros

A aplicação de outras metaheurísticas tanto na fase de exploração do algoritmo *Q-Learning* como na fase de explotação é sugerida como trabalhos futuros.

6.4 Produção Científica

PAIVA, É. O.; GUELPELI, M. V. C. Improvement of Q-Learning Algorithm Convergence with Intelligent Tutoring Systems and Tabu Search. International Journal of Modern Education Research, v. 3, p. 1-5, 2016.
Disponível em: < http://www.aascit.org/journal/archive2?journalId=910&paperId=3525>.

REFERÊNCIAS

BELLMAN, R. **Dynamic Programming**. Princeton: Princeton University Press, 1957.

BIANCHI, R. A. C.; COSTA, A. H. R. **Uso de heurísticas para a aceleração do aprendizado por reforço**. XXV Congresso da Sociedade Brasileira de Computação, 2005.

CALLEGARI-JACQUES, S. M. **Bioestatística: Princípios e Aplicações**. Porto Alegre: Artmed, p. 264, 2007.

DEEPHI, P.; SASIKUMAR, M. **Student model for an intelligent language tutoring system**. IEEE 14th International Conference on Advanced Learning Technologies (ICALT), p. 441 - 443, 2014.

FEO, T. A.; RESENDE, M. G. C. **Greedy Randomized Adaptive Search Procedures. Journal of Global of Optimization**, p. 106 - 134, 1995.

GAVIDIA, J. J. Z.; ANDRADE, L. C. V. **Sistemas Tutores Inteligentes.** 2003. Trabalho de conclusão da disciplina Inteligência Artificial, Programa de Pós-Graduação da COPPE, Universidade Federal do Rio de Janeiro. Rio de Janeiro, 2003.

GIRAFFA, L. M. M. **Uma arquitetura de tutor utilizando estados mentais**, 1999.

GLOVER, F. **Future paths for integer programming and links to artificial intelligence**. Computers an Operations Research, 1986. 533 - 549.

GLOVER, F.; KOCHEMBERGER, G. A. **Handbook of Metaheuristics**. Boston: Kluwer Academic, 2003.

GLOVER, F.; LAGUNA, M. **Tabu search**. Kluwer Academic Publishers, 1997.

GUELPELI, M. V. C. et al. **The aprendice modeling through reinforcement with a temporal analysis using Q-Learning algorithm**. International Conference on Computer Science and Automation Engennering, 2012.

GUELPELI, M. V. C.; OMAR, N.; RIBEIRO, C. H. C. **Aprendizado por reforço para um sistema tutor inteligente sem modelo explícito do aprendiz**. Revista Brasileira de Informática na Educação, v. 12, p. 69 – 77, 2004. ISSN 2.

JAVADI, S. L.; MASOUMI, B.; MEYBODI, M. R. **Improving student's modeling framework in a tutorial-like system based on pursuit learning automata and reinforcement learning**. International Conference on Education and e-Learning Innovations, 2012.

JESUS, A. **Sistema Tutores Inteligentes: Uma Visão Geral**. Revista Eletrônica de Sistemas de Informação, ISSN 1677-3071 doi:10.5329/RESI, 2009.

JHONSON, D. S.; MCGEOCH, L. A. **Local Search in Combinatorial Optimization**. [S.l.]: Princeton University Press, 2003.

LI, D.; ZHOU, H. H. **An intelligent tutoring system with an automated knowledge acquisition mechanism**. IEEE International Conference on Computational Intelligence & Communication Technology, 2015.

LITTMAN, M. L.; SZEPESVARI, C. A **Genreralized Reinforcement-Learning Model**, 1996.

LUGER, G. F. **Inteligência Artificial**. 6. ed. São Paulo: Pearson, 2013.

NORVING, P.; RUSSELL, S. **Inteligência Artificial**. 3. ed. [S.l.]: Elsevier, 2013.

PAIVA, É. O.; GUELPELI, M. V. C. **Improvement of Q-Learning Algorithm Convergence with Intelligent Tutoring Systems and Tabu Search**. International Journal of Modern Education Research, v. 3, p. 1-5, 2016. Disponível em: < http://www.aascit.org/journal/archive2?journalId=910&paperId=3525>.

PALOMINO, C. E. G. **Modelo de Sistema Tutorial Inteligente para Ambientes Virtuais de Aprendizagem baseado em Agentes**, Florianópolis, 2013.

RIBEIRO, C. C.; RESENDE, M. G. C. **Greedy Randomized Adaptive Search Procedures**. Kluwer Academic Publishers, p. 219 - 249, 2003.

SANTOS, J. P. Q. **Uma implementação paralela híbrida para o problema do caixeiro viajante usando algotítmos genéticos, GRASP e aprendizado por reforço**, 2009.

SUTTON, R. S.; BARTO, A. G. **Reinforcement Learning: An Introduction**. [S.l.]: An Bradford Book, 1998.

TURING, A. M. **Computing machinery and intelligence**. Mind, n. 49, p. 433 - 460, 1950.

VICCARI, R. M. **Um tutor inteligente para a programação em lógica: idealização, projecto e desenvolvimento**. Coimbra: [s.n.], 1989.

WATKINS, C. J. H. **Learning from delayed rewards: Convergence and applications**, 1989.

ZHANG, X.; LIU, Z. **An optimized q-learning algorithm based on the thinking of tabu search**. International Symposium on Computational Intelligence and Design, p. 533 - 536, 2008. Disponivel em: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4725666>.

APÊNDICES

APÊNDICE A

Telas do Simulador

O Apêndice A apresenta as telas do simulador utilizado nas simulações deste trabalho. O programa é dividido em três telas: Configuração, Simulação e Visita aos Estados.

A Figura 32 apresenta as a tela do simulador responsável pelas opções de configurações das simulações:

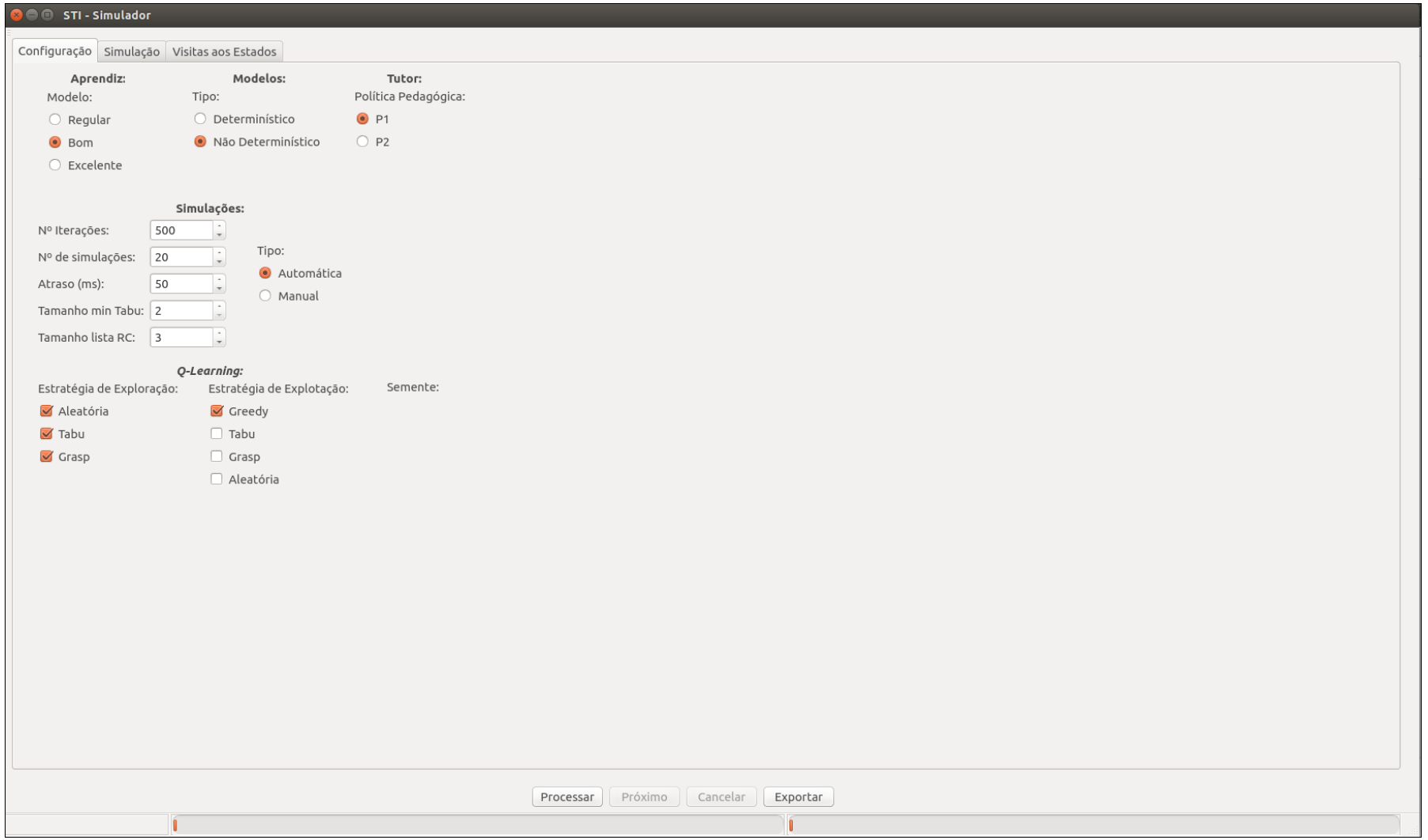

Figura 32 – Simulador – Tela Configuração

A Figura 33 apresenta os resultados parciais das simulações em tempo real:

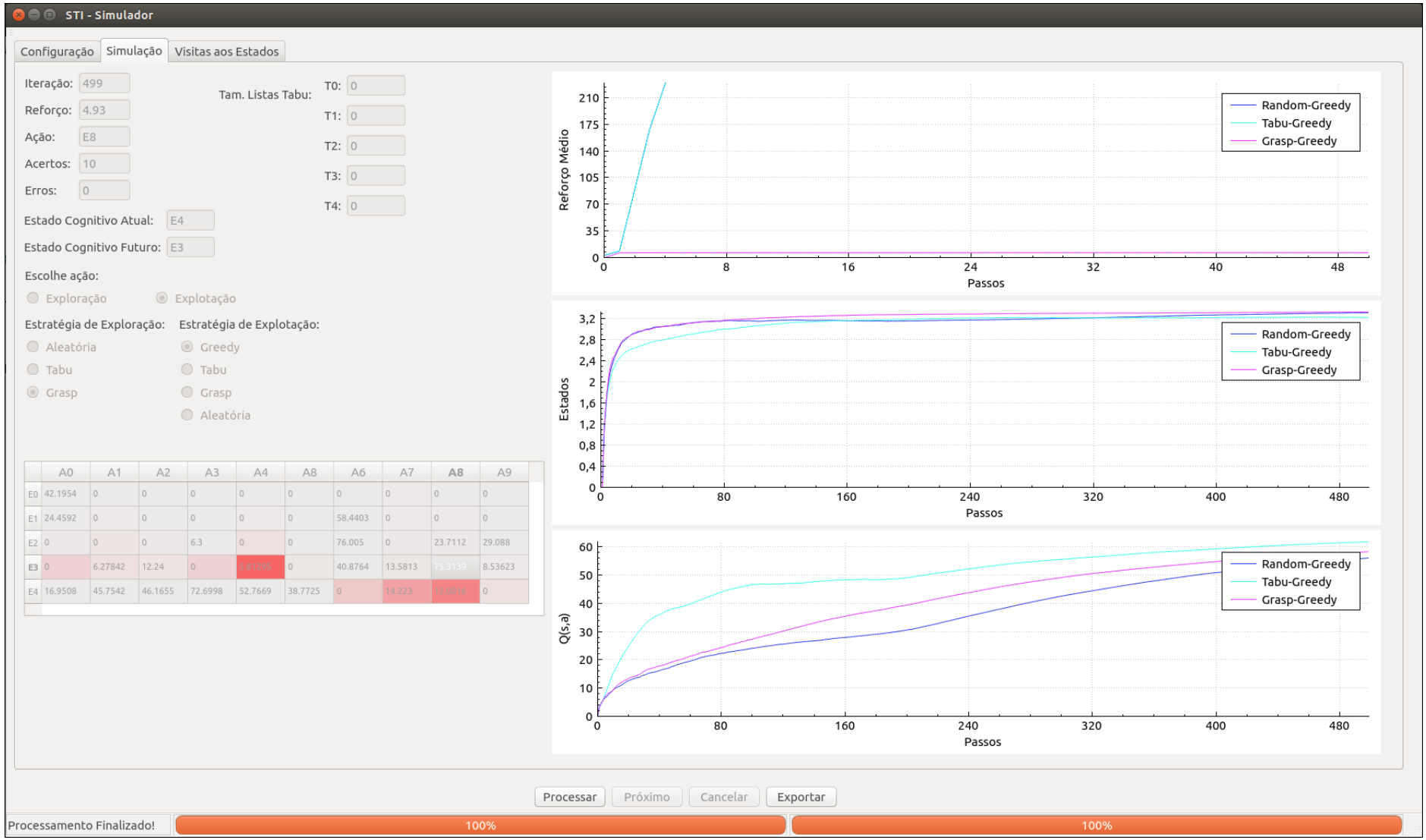

Figura 33 – Simulador – Tela Simulação

A Figura 34 o percentual de visita aos estados obtido para cada uma das configurações de simulação:

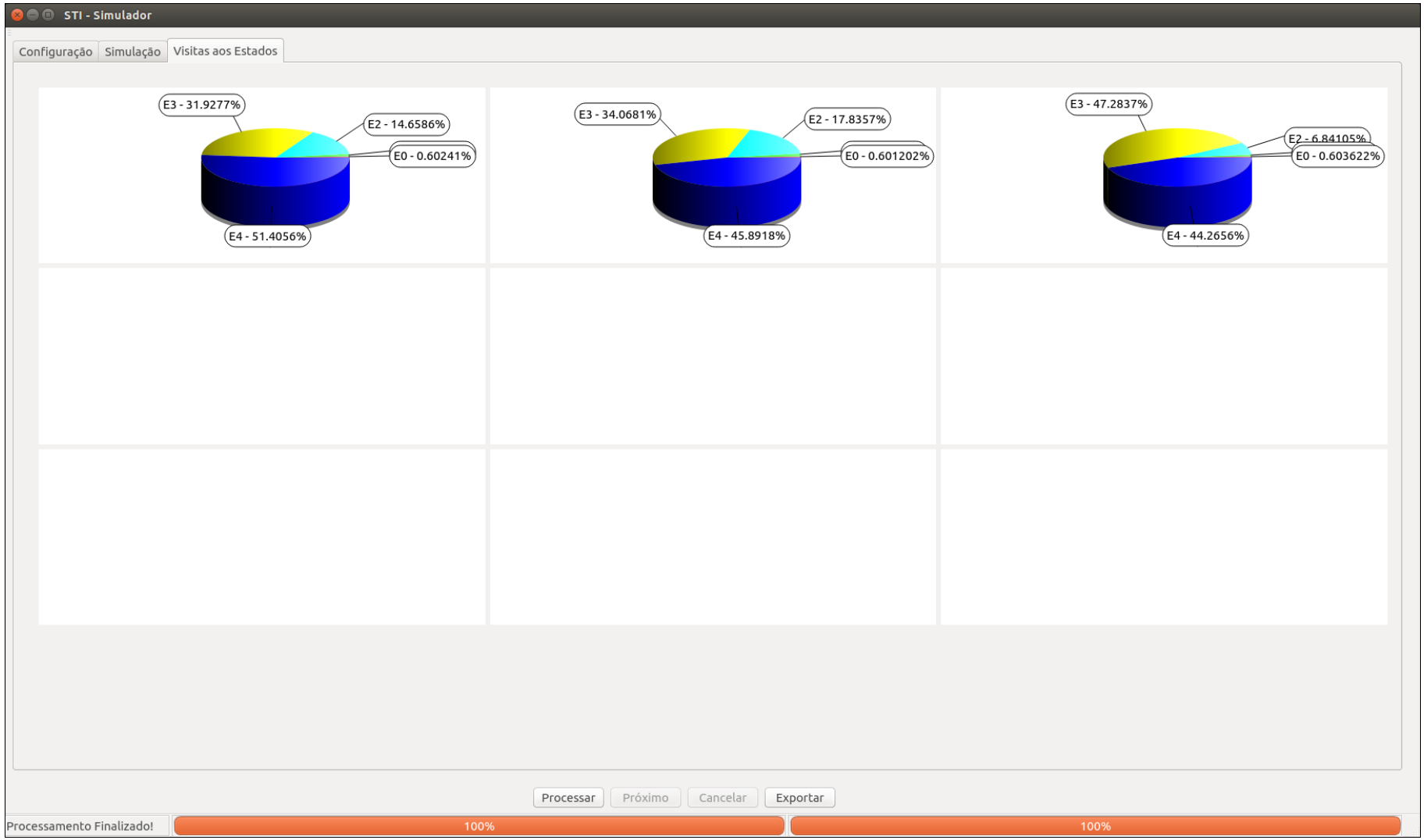

Figura 34 – Simulador – Tela Visita aos Estados

APÊNDICE B

Valor de Utilidade da Ação ($Q(s, a)$) – Tamanho da Lista Tabu

O Apêndice B mostra os resultados de desempenho comparativo do valor de utilidade $Q(s, a)$ de um par (estado(s), ação(a)) da heurística Busca Tabu aplicada como estratégia de exploração de modelo de aprendiz **M2** – **Bom**. As simulações foram aplicadas em **500** passos com o objetivo de identificar qual valor para o parâmetro tamanho da lista Tabu obterá o melhor desempenho. Foram testados os valores 2, 3, 5 e 7 para o parâmetro **tamanho da lista Tabu**. Os resultados apresentados foram obtidos através da média de 20 execuções de cada simulação.

Modelo M2 – Bom – 500 Passos – Tamanho da Lista Tabu

Disponível eletronicamente em: https://goo.gl/wdzuyQ

Planilha: 01 – TamanhoListaTabu.

APÊNDICE C

Valor de Utilidade da Ação ($Q(s, a)$) – Tamanho da Lista Restrita de Candidatos

O Apêndice C mostra os resultados de desempenho comparativo do valor de utilidade $Q(s, a)$ de um par (estado(s), ação(a)) da heurística Busca Tabu aplicada como estratégia de exploração de modelo de aprendiz **M2** – **Bom**. As simulações foram aplicadas em **500** passos com o objetivo de identificar qual valor para o parâmetro tamanho da lista Restrita de Candidatos obterá o melhor desempenho. Foram testados os valores 2, 3, 4 e 5 para o parâmetro **tamanho da lista Restrita de Candidatos**. Os resultados apresentados foram obtidos através da média de 20 execuções de cada simulação.

Modelo M2 – Bom – 500 Passos – Tamanho da Lista Restrita de Candidatos

Disponível eletronicamente em: https://goo.gl/wdzuyQ

Planilha: 02 - TamanhoLRC.

APÊNDICE D

Valor de Utilidade da Ação ($\boldsymbol{Q(s,a)}$)

O Apêndice D mostra os resultados de desempenho do valor de utilidade $Q(s,a)$ de um par (estado(s), ação(a)) das heurísticas Aleatória, Tabu e GRASP aplicadas como estratégia de exploração de aprendiz **M1** – **Ruim**, **M2** – **Bom** e **M3** – **Excelente**. Para cada um desses modelos de aprendiz foram realizadas simulações de **300**, **500** e **1000** passos. Os resultados apresentados foram obtidos através da média de 20 execuções de cada simulação.

Modelo – Passos	**Disponível eletronicamente em: https://goo.gl/wdzuyQ**
Modelo M1 – Ruim – 300 Passos	Planilha: 03 - M1-300
Modelo M1 – Ruim – 500 Passos	Planilha: 04 – M1-500
Modelo M1 – Ruim – 1000 Passos	Planilha: 05 - M1-1000
Modelo M2 – Bom – 300 Passos	Planilha: 06 – M2-300
Modelo M2 – Bom – 500 Passos	Planilha: 07 – M2-500
Modelo M2 – Bom – 1000 Passos	Planilha: 08 – M2-1000
Modelo M3 – Excelente – 300 Passos	Planilha: 09 – M3-300
Modelo M3 – Excelente – 500 Passos	Planilha: 10 – M3-500
Modelo M3 – Excelente – 1000 Passos	Planilha: 11 – M3-1000