

## Um Arcabouço Adaptativo para Objetos de Aprendizagem

Rômulo C. Silva<sup>1,2</sup>, Alexandre I. Direne<sup>2</sup>, Diego Marczał<sup>3</sup>, Ana Carla Borille<sup>2</sup>  
Paulo R. B. Guimarães<sup>4</sup>, Ângelo S. Cabral<sup>4</sup>, Bruno F. Camargo<sup>4</sup>

<sup>1</sup>Centro de Engenharia e Ciências Exatas (CECE)  
Universidade Estadual do Oeste do Paraná (UNIOESTE)  
Foz do Iguaçu – PR

<sup>2</sup>Departamento de Informática – Universidade Federal do Paraná (UFPR)  
Curitiba – PR

<sup>3</sup>Universidade Tecnológica Federal do Paraná (UTFPR)  
Guarapuava – PR

<sup>4</sup>Departamento de Estatística – Universidade Federal do Paraná (UFPR)  
Curitiba – PR

{romulo cesarsilva, dmarczal, carlaborille}@gmail.com,

alexnd@inf.ufpr.br,

{guimaraes.prb, angeloscabral}@gmail.com,

brunofilla.camargo@hotmail.com

**Abstract.** *The work approaches theoretical and implementation issues of a framework for creating and executing Learning Objects (LOs) where problem-solving tasks are ordered according to the matching of two parameters: (1) student skill level and (2) problem solution difficulty. Both are formally defined as algebraic expressions. The definition of skill level is achieved through a rating-based measure that resembles the ones of game mastery scales, while the solution difficulty is based on mistakes and successes of learners to deal with the problem. Also, the results of an experiment conducted with four highschool classes using the framework for the domain of logarithmic properties are presented.*

**Resumo.** *Este trabalho aborda questões teóricas e de implementação de um arcabouço para a construção e execução de Objetos de Aprendizagem (OAs), em que as tarefas de resolução de problemas são ordenadas de acordo com o emparelhamento de 2 parâmetros formalmente definidos como expressões algébricas: (1) nível de expertise do aluno e (2) dificuldade de solução de um problema. O nível de habilidade é expresso por um rating, semelhante aos usados em jogos. O cálculo da dificuldade de solução é baseado em erros e acertos de estudantes ao lidar com o problema. São apresentados resultados de um experimento realizado com quatro turmas do ensino médio de uma escola pública, utilizando um OA construído com o arcabouço para o domínio de logaritmos.*

### 1. Introdução

O expertise do estudante é geralmente desenvolvido através da resolução de exercícios que requerem um conjunto de habilidades avaliadas. Isso é feito tanto no sistema edu-

cacional de sala de aula convencional quanto em sistemas que fazem uso de computadores e aplicam técnicas de Inteligência Artificial tais como Sistemas Tutores Inteligentes (STIs). Normalmente, os professores detectam equívocos conceituais dos alunos durante a realização de testes e exercícios. Dependendo de como a solução dada diverge da resposta correta ou solução esperada, dois estudantes que erraram a mesma questão podem ser pontuados diferentemente para aquela questão específica.

Outra característica que pode ser usada para compor a nota é o grau de dificuldade da questão, podendo ser medido pelo número de estudantes que a saltaram ou erraram. Assim, um estudante que responde corretamente uma questão que muitos erraram, provavelmente tem mais capacidade no assunto avaliado que os demais e sua nota deve refletir isso. Inversamente, um estudante que erra uma questão que vários outros acertam, possivelmente tem menos capacidade no tópico avaliado. Este trabalho aborda questões teóricas e de implementação de um arcabouço para a construção e execução de Objetos de Aprendizagem inteligentes destinados a reconhecer o nível de competência do aprendiz e recomendar exercícios em grau de dificuldade compatível.

Um aspecto importante associado à modelagem do estudante é a avaliação correta de sua perícia visando a calibragem do grau de dificuldade do conteúdo e a sugestão de exercícios de acordo com o perfil do aprendiz. Além disso, considerando os aspectos motivacionais em aprendizagem, é importante que o STI seja capaz de sugerir atividades adequadas ao nível de expertise do estudante. Tal situação se justifica pois, a apresentação de exercícios com grau de dificuldade muito abaixo do nível cognitivo do aprendiz pode causar entediamento enquanto exercícios muito acima podem gerar desmotivação. Ambas as situações podem gerar o abandono da atividade proposta.

Em muitas visões da pesquisa de STIs, como na de Self [Self 1990], a criação de modelos precisos do aprendiz permitem atribuir ao aprendiz atividades práticas baseadas em avaliações de seu conhecimento pessoal. Além disso, modelos de estudante tem se tornado um elemento chave em STIs, apoiando o desenvolvimento de ajuda individualizada e detecção de comportamento desviado da tarefa proposta [Baker et al. 2010]. As abordagens mais recentes de comportamento deslocado do esperado são influenciadas pelo comportamento de outros estudantes. Nesse sentido, uma amostragem maior de dados dos alunos pode prover melhores avaliações automáticas de um aprendiz específico.

Um erro do estudante pode basicamente ser usado para duas ações: simplesmente avaliar o estudante ou detectar equívocos conceituais visando uma prática pedagógica mais efetiva. Nessa última perspectiva, recentemente tem crescido o interesse pelo uso direto de erros como fonte de material didático, como por exemplo a técnica de exemplos errôneos. Essa técnica consiste de uma descrição de como solucionar um problema no qual um ou mais passos estão incorretos, sendo pedido ao estudante para encontrá-lo(s), explicar ou corrigir o(s) erro(s) visando fazê-lo aprender mais profundamente sobre o conteúdo do domínio e desenvolver habilidades metacognitivas [Isotani et al. 2011].

## **2. Trabalhos Relacionados**

Pinkwart e Loll [Pinkwart and Loll 2009] apresentaram um estudo comparativo entre três abordagens de avaliação da qualidade das soluções dos estudantes, sendo que duas fazem uso de revisões por pares: a qualidade da solução de uma tarefa de um estudante é determinada heurísticamente por avaliações de outros estudantes. Na primeira abordagem,

quando um estudante trabalha em uma tarefa e provê uma solução, depois é solicitado a ele avaliar algumas soluções alternativas. Uma avaliação terá um peso maior que outras quando o estudante que a faz tem um *rating* de qualidade mais alto quando comparado a outros. A segunda abordagem assume que um estudante que consegue classificar corretamente a qualidade de soluções alternativas, também está apto a prover uma solução de maior qualidade. Enquanto a primeira faz um uso clássico de revisão por pares com um valor base fixo para aquelas soluções que não foram avaliadas ainda, a segunda substitui este valor base por um *rating* calculado dinamicamente. A terceira abordagem usa somente a fórmula de *rating* base. O processo de colaboração entre os usuários pode ser classificado como uma aplicação de Web Semântica Social, que tem o potencial de diminuir a sobrecarga dos tutores e prover a possibilidade dos estudantes treinarem suas capacidades de criticidade.

Ravi e Sosnovsky [Ravi and Sosnovsky 2013] propõem um método de calibragem para dificuldade de solução em STIs baseado na aplicação de técnicas de mineração de dados no log de interação do estudante com o sistema. Usando o método bayesiano clássico *Knowledge Tracing* (KT) [Corbett and Anderson 1994], a probabilidade que um estudante tenha adquirido uma habilidade é calculada com base na sequência de tentativas de exercícios para os quais as soluções envolvem um dado conceito. Os eventos no *log* são agrupados por exercícios e classificados de acordo com a habilidade do estudante. Os dados gerados são usados para se encaixarem na curva sigmoide da TRI (Teoria da Resposta ao Item)[Baker 2001] para conectar estudantes diferentes usando o algoritmo de *clustering* padrão *k-means*.

Champaign e Cohen propõem um algoritmo [Champaign and Cohen 2010] para sequenciamento de conteúdo que seleciona o objeto de aprendizagem (OA) apropriado a ser apresentado a um estudante, baseado em experiências de aprendizagem prévias de usuários similares. A granulação de sequenciamento ocorre no nível do OA, não de exercícios ou questões. Uma limitação do trabalho é que o algoritmo foi validado usando somente estudantes simulados.

Schatten e Schmidt-Thieme [Schatten and Schmidt-Thieme 2014] apresentam o Vygotski Policy Sequencer (VPS), baseado no conceito de Zona de Desenvolvimento Proximal (ZDP) idealizado por Vygotski. Nessa abordagem, a matriz de fatoração, que é um método para predição de *rating* de usuário, é combinada com uma política de sequenciamento, visando selecionar a cada passo o conteúdo de acordo com o escore estimado. O VPS é constituído de tal forma que a definição explícita de dificuldade não é necessária. Os autores também mostram como usar estudantes simulados para testar as sequências antes de serem aplicadas a estudantes reais.

Segal *et al.* [Segal et al. 2014] propõem um algoritmo denominado EduRank para personalizar conteúdo educacional para estudantes em sistemas *e-learning*, baseado em técnicas de filtragem colaborativa combinadas com a teoria da escolha social. EduRank constrói um *ranking* de dificuldade das questões para um estudante particular baseado na agregação do *ranking* de estudantes similares, medido por diferentes aspectos de seu desempenho em questões comuns passadas, tais como notas, número de tentativas e tempo gasto resolvendo questões.

### 3. Conceitos da Solução

Este estudo propõe um método para calibragem de exercícios e um algoritmo associado para seu sequenciamento em OAs. Adicionalmente, também é apresentada uma fórmula para classificar a perícia do aluno, inspirada em sistemas de *rating* de jogos adversaristas como xadrez. O *rating* é um número real em um intervalo  $[minRank, maxRank]$  tal que é muito improvável um jogador (ou aluno) atingir os extremos do intervalo.

#### 3.1. Cálculo Automático de *Rating*

As seguintes diretrizes foram adotadas para a elaboração das fórmulas que definem o *rating* do aluno e grau de dificuldade das questões:

- a cada questão é atribuído um grau de dificuldade no intervalo  $[0..10]$ ;
- similarmente, cada estudante tem seu *rating* no intervalo  $[1..10]$  para expressar sua perícia no assunto em questão;
- quanto mais fácil uma questão, maior a probabilidade que os estudantes a respondam corretamente. Nesse caso, o *rating* de um estudante terá um pequeno aumento caso acerte a questão ou maior diminuição no caso de erro;
- estudantes que têm sucesso na primeira tentativa de solução de uma questão são pontuados com uma nota mais alta no seu *rating* quando comparados com outros que precisaram de várias tentativas;
- questões saltadas são consideradas erradas.

Este estudo propõe a Equação 1 para avaliar iterativamente a habilidade de um estudante, e considera os acertos, os erros e o número de tentativas, tanto do próprio aluno quanto dos demais, como se houvesse implicitamente uma competição entre todos os alunos da turma.

$$R_J^q = R_J^{q-1} + Ak_1\alpha\left(10 - \frac{9T_J^q}{T_{med}^q}\right) - Ek_2\beta \times 10\frac{T_J^q}{T_{med}^q} \quad (1)$$

- $R_J^q$ : *rating* do estudante  $J$  após responder a questão  $q$ ;
- $A = 1$  e  $E = 0$  se o estudante acertou  $q$ , caso contrário  $A = 0$  e  $E = 1$ ;
- $T_J^q$ : número de tentativas sem sucesso do estudante  $J$  ao responder a questão  $q$ ;
- $T_{med}^q$ : mediana do número de tentativas erradas da questão  $q$  durante a sessão de exercícios;
- $N_a^q$ : número de estudantes que tiveram sucesso em responder a questão  $q$ ;
- $N_e^q$ : número de estudantes que erraram a questão  $q$ ;
- $\alpha = \frac{1}{N_a^q}$ : ponderação do incremento de *rating*;
- $\beta = \frac{1}{N_e^q}$ : ponderação do decremento de *rating*;
- $k_1$  e  $k_2$ : fatores multiplicadores incremento e decremento de *rating*, respectivamente, calculados de acordo com  $R_J^{q-1}$  tal que  $1 \leq R_J^{q-1} \leq 10$ , and  $k_1 = 1 - \frac{R_J^{q-1}}{10}$  e  $k_2 = \frac{R_J^{q-1} - 1}{10}$ ;
- $10 - \frac{9T_J^q}{T_{med}^q}$ : nota na questão  $q$  em caso de acerto;
- $10\frac{T_J^q}{T_{med}^q}$ : nota na questão  $q$  em caso de erro.

Além disso, não existe limite para o número de tentativas. Entretanto, se há mais de 10 tentativas, 10 é considerado para propósitos de cálculo. Os fatores  $k_1$  e  $k_2$  evitam que o resultado da Equação 1 atinja os extremos do intervalo [1..10]. O *rating* inicial  $R_j^0$  é 5.5 (ponto médio do intervalo de *rating*) para todos estudantes inicialmente, quando não se tem qualquer informação a seu respeito.

### 3.2. Grau de Dificuldade

Em uma primeira abordagem, o grau de dificuldade da questão  $q$  é definido através dos detalhes da Equação 2 e seus parâmetros que se seguem:

$$D^q = \frac{N_e^q + N_s^q}{N_e^q + N_a^q + N_s^q} \times 10 \quad (2)$$

onde:

- $D^q$ : grau de dificuldade da questão  $q$  após uma sessão de exercícios;
- $N_e^q$ : número de estudantes que erraram  $q$ ;
- $N_a^q$ : número de estudantes que acertaram  $q$ ;
- $N_s^q$ : número de estudantes que saltaram  $q$ .

Em outra abordagem, usando somente o número de tentativas e considerando que o estudante geralmente tenta até conseguir obter a resposta correta, o grau de dificuldade de uma questão  $q$  pode ser definido pela Equação 3 e seus parâmetros que seguem:

$$D^q = \frac{\sum_{J=0}^{J=n} T_J^q}{N_e^q + N_a^q} \quad (3)$$

onde:

- $D^q$ : grau de dificuldade da questão  $q$  após uma sessão de exercícios;
- $T_J^q$ : número de tentativas do estudante  $J$  na questão  $q$ . Se o número de tentativas é maior que 10, então 10 é considerado como  $T_J^q$ ;
- $N_e^q$ : número de estudantes que erraram  $q$ ;
- $N_a^q$ : número de estudantes que acertaram  $q$ .

Nesse caso, quanto mais difícil a questão, maior a probabilidade de que os estudantes precisem de mais tentativas para chegar à resposta correta. Além disso, o número de estudantes que saltaram  $q$  não é considerado.

### 3.3. Algoritmo para Sequenciamento Adaptativo de Exercícios

Um aspecto importante em STIs é como os exercícios devem ser sequenciados após serem calibrados tal que se encaixem com o nível de perícia do estudante. No início, o sistema não tem qualquer informação a respeito do estudante. Portanto, uma maneira natural é colocá-los em ordem crescente de dificuldade. Entretanto, existem muitas questões em aberto relacionadas ao número de exercícios que devem ser apresentados, e quantas vezes.

Neste trabalho, é proposto um algoritmo para sequenciamento de exercícios em ordem crescente de dificuldade, combinado com um mecanismo similar à interpolação numérica, que segue as diretrizes abaixo:

- uma sequência mínima de exercícios é definida, que será apresentada inicialmente ao estudante;
- a sequência mínima sempre começa com o exercício mais fácil e termina com o mais difícil;
- os exercícios de dificuldade intermediária na sequência mínima são distribuídos uniformemente entre o mais fácil e o mais difícil tal que o número de exercícios é  $\left\lceil \frac{n}{stepsize} \right\rceil$  onde  $n$  é o total de exercícios e  $stepsize$  refere-se ao número de exercícios que podem ser saltados quando o estudante acerta o exercício;
- o número de tentativas que um estudante tem para resolver uma questão é limitado à média de tentativas obtidas durante a fase de calibragem;
- quando o número de tentativas excede o limite estabelecido, o próximo exercício a ser apresentado ao estudante é o que se encontra no ponto médio de dificuldade entre o exercício atual e o último respondido corretamente.

Por exemplo, considere um OA com 30 exercícios em ordem crescente de dificuldade  $[e_1, e_2, \dots, e_{30}]$  e  $stepsize = 4$ . A sequência mínima de exercícios será  $min\_seq = \langle e_1, e_5, e_9, e_{13}, e_{17}, e_{21}, e_{25}, e_{29}, e_{30} \rangle$ , e os exercícios serão apresentados ao estudante considerando inicialmente esta ordem. Porém, por exemplo, se o estudante excede o número de tentativas estabelecido para  $e_9$ , então  $e_7$  (ponto médio entre  $e_5$  e  $e_9$ ) é apresentado. Ao contrário do que ocorre durante a fase de calibragem dos exercícios, o(a) estudante não pode saltar exercícios. Além disso, se ele/ela erra continuamente, a apresentação dos exercícios passa a ser sequencial. O  $stepsize$  pode ser configurado pelo autor do OA e deve ser tal que o número de exercícios na sequência mínima tenha pelo menos 25% do total de exercícios.

Na Figura 1 é apresentada uma árvore de sequenciamento de exercícios considerando  $stepsize = 4$ . Nós-filhos à direita representam o próximo exercício quando o estudante respondeu corretamente ao exercício correspondente ao nó-pai. Enquanto o estudante erra um exercício  $e_t$ , o mesmo é apresentado novamente até o limite de tentativas. Após isso, o exercício representado pelo nó-filho à esquerda é exibido.

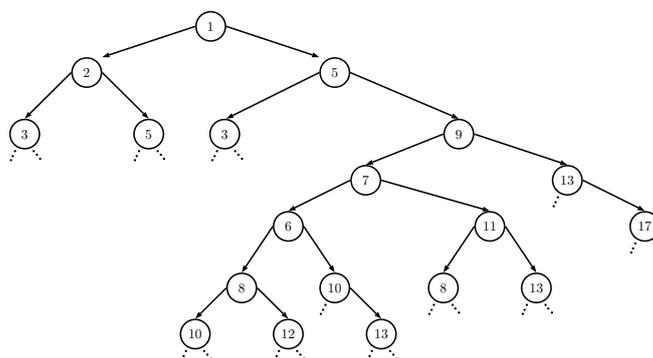


Figura 1. Árvore de Sequenciamento de Exercícios

#### 4. O Software ADAPTFARMA

ADAPTFARMA é uma versão modificada de FARMA (Ferramenta de Autoria para a Remediação de erros com Mobilidade na Aprendizagem)[Marczal and Direne 2012,

Marczal 2014], uma ferramenta de autoria para construção de OAs de conceitos matemáticos. Do mesmo modo que em FARMA, em ADAPTFARMA um OA consiste de uma sequência de exercícios após uma introdução. A introdução é a parte teórica de um OA onde conceitos são definidos através de texto, imagens, audios e vídeos. Os exercícios contêm a essência prática dos conceitos, com os quais o aprendiz irá adquirir perícia através de tarefas de resolução de problemas. Adicionalmente, ADAPTFARMA tem integrado em seu arcabouço a implementação do cálculo de *rating* dos alunos, o grau de dificuldades das questões e o algoritmo de sequenciamento adaptativo de exercícios apresentados na seção 3.

Para construir uma introdução e seus exercícios correspondentes, ADAPTFARMA oferece uma interface WYSIWYG (*What You See Is What You Get*), similar aos processadores de texto altamente interativos. Em relação aos exercícios, a ferramenta é bastante genérica, permitindo ao professor definir o número de questões relacionadas a cada exercício. Esta técnica é baseada no teoria ACT (Adaptive Control of Thought)[Ritter et al. 2007]. Para cada questão, o professor-autor deve indicar uma solução de referência, que é a resposta correta da questão. ADAPTFARMA permite que expressões algébricas e aritméticas sejam colocadas como solução de referência. Sob o modo estudante, a ferramenta lida automaticamente com equivalência entre a resposta do estudante e a solução de referência.

Além disso, ADAPTFARMA permite ao professor-autor definir regras de remediação e mensagens para cada questão dos exercícios, aplicadas quando o estudante comete um erro. Para tal, é necessário definir o número de tentativas sem sucesso após o qual a mensagem de *feedback* deve ser exibida. Isto mostra o quanto é simples proporcionar *feedback* imediato em ADAPTFARMA.

A capacidade de retroagir ao contexto exato do erro cometido pelo estudante é uma funcionalidade diferenciada de ADAPTFARMA, quando comparada a ferramentas similares, pois dá oportunidade ao professor de identificar os passos equivocados feitos pelo estudante para então lidar apropriadamente com as causas do erro. Além disso, ADAPTFARMA permite ao professor ver uma interação completa do estudante com a ferramenta em ordem cronológica, na forma de uma *timeline*. Da mesma maneira, os aprendizes também podem retroagir ao contexto exato de qualquer uma das suas respostas corretas ou erradas, visando refletir sobre seus próprios passos da solução.

ADAPTFARMA é arquiteturalmente dividida em 3 módulos principais: autoria, interação e monitoramento. O módulo de autoria provê as funcionalidades para construção de OAs. O módulo de interação é a interface entre o aprendiz e o OA criado pelo professor, efetuando a junção das introduções, exercícios, questões e remediação de erros, para executar o OA como uma sequência de páginas.

## 5. Avaliação Experimental

Para avaliar a efetividade da aprendizagem associada às estratégias de sequenciamento, foi realizado um experimento com quatro turmas de estudantes do Ensino Médio de uma escola pública. O mesmo OA sobre o domínio de logaritmos, criado usando o ambiente ADAPTFARMA e contendo 30 exercícios, foi aplicado às quatro turmas. Para cada turma, o OA foi aplicado com um sequenciamento diferente para os exercícios:

- turma A: método de sequenciamento randômico (MSR);

- turma B: método de sequenciamento definido pelo professor (MSP);
- turma C: método de sequenciamento em ordem crescente de dificuldade (MSD), onde o grau de dificuldade foi calculado segundo a Equação 3 definida na Seção 3.2 usando os dados obtidos da fase de calibragem da turma A;
- turma D: método de sequenciamento adaptativo (MSA), usando o algoritmo descrito na Seção 3.3.

Os mesmos pré e pós-testes foram aplicados a todas as turmas, valendo 1, 5 cada. Os alunos que não participaram de qualquer uma das etapas foram excluídos da análise, resultando em 119 participantes. Para os métodos MSR, MSP e MSD não havia limite para o número de tentativas, enquanto para o MSA a média de tentativas da turma A foi utilizada. Na Tabela 1 são mostrados o número de alunos participantes do experimento, a média das notas no pré- e no pós-teste de cada turma.

**Tabela 1. Médias do Pré-teste e Pós-teste**

Turma	Número de Alunos	Média no Pré-teste	Média no Pós-teste
A	29	1.0414	1.2172
B	30	0.9067	1.1100
C	30	0.9000	0.9233
D	30	0.9867	1.1267

O teste Shapiro-Wilk foi aplicado às notas obtidas para checar normalidade. Como somente os dados do MSD passaram no teste de normalidade ( $p$ -value = 0.0827), então o Teste T pareado foi aplicado a essa turma, enquanto para as outras três, foi aplicado o teste Wilcoxon para avaliar individualmente cada método de sequenciamento. Os  $p$ -valores resultantes foram: MSR(0.0007), MSP(< 0.0001), MSD(0.5302) e MSA (0.0037), mostrando que com exceção de MSD, todos os métodos tiveram um aumento significativo nos escores entre o pré- e o pós-teste.

O método ANOVA foi aplicado aos dados do pré-teste que exibiram normalidade, obtendo  $p$ -value = 0.2539, demonstrando não haver diferença significativa entre as 4 turmas. Para os dados do pós-teste e diferença média entre pré e pós-testes foi aplicado o teste Kruskal-Wallis, obtendo  $p$ -valor = 0.00579 e  $p$ -valor = 0.03073 respectivamente, demonstrando existir diferença significativa e sugerindo que MSR, MSP e MSA levam o estudante a obter uma melhor performance que MSD. Além disso, a performance entre MSR, MSP e MSA foram similares. Supreendentemente, MSR obteve a melhor performance enquanto MSD a pior. Isto contradiz um grande número de pesquisas referenciadas na literatura sobre práticas pedagógicas usando computadores [Direne 1990] ou por outro lado, para desenvolvimento de habilidades de solução de problemas. Algumas razões que podem explicar esse fenômeno são:

- o problema da ordenação de exercícios é uma questão relevante que deve ser explorada mais detalhadamente para permitir a verificação da influência tácita do conhecimento contido na organização textual do problema;
- a falta de diferença significativa entre MSR, MSP e MSA é também confirmada por evidências achadas em pesquisas passadas, como no experimento descrito em [Major and Reichgelt 1992, Mitrovic 2003];

- o MSD pode ter se ligado a algum tópico relevante que causou um aumento da carga cognitiva, resultando em soluções de problemas que divergem das corretas.
- embora a maioria dos estudantes tenham participado do experimento, somente as notas do pré- e pós-teste foram levadas em conta na nota final constante no registro acadêmico do estudante.

## 6. Conclusão e Trabalhos Futuros

Geralmente a perícia do estudante é desenvolvida pela solução de exercícios que requerem um conjunto de habilidades avaliadas, inclusive em STIs. Este trabalho propôs um sistema de *rating* automático que pode ser usado como ferramenta adicional de avaliação dos estudantes. Dependendo do número de tentativas e o grau de dificuldade de uma questão, estudantes podem obter notas diferentes para a mesma questão. Além disso, duas fórmulas para o cálculo do grau de dificuldade de uma questão foram apresentadas.

O sistema de *rating* implicitamente estabelece um *ranking* entre os estudantes que pode auxiliar os professores identificarem quem necessita mais assistência. Além disso, ele permite que o professor obtenha um *feedback* sobre o grau de dificuldade das questões que elabora, e que essa informação possa ser agregada em repositórios de exercícios, habilitando futuramente a montagem automática de OAs.

Este estudo também propôs um algoritmo para sequenciamento de exercícios que usa o grau de dificuldade combinado com um mecanismo similar à interpolação numérica. O algoritmo foi implementado para compor o ambiente ADAPTFARMA, uma ferramenta de autoria web com interface WYSIWYG para criação e execução de OAs. O ambiente ADAPTFARMA está implementado de tal forma que é muito simples alterar a estratégia de sequenciamento de exercícios. Aproveitando dessa característica, foi realizado um experimento com quatro turmas do Ensino Médio de uma escola pública, para testar as seguintes estratégias de sequenciamento: MSR, MSP, MSD e MSA. Surpreendentemente somente MSD não obteve um aumento significativo nas notas dos alunos e MSR obteve a melhor performance, demonstrando que o problema da ordenação de exercícios é uma questão relevante, devendo ser pesquisada mais cuidadosamente.

Os trabalhos futuros serão concentrados em adicionar funcionalidades ao ambiente ADAPTFARMA, que atuem de duas maneiras. Primeiro desenvolvendo uma abordagem mais profunda de adaptação ao usuário que inclua mais dimensões além da unificação entre a dificuldade do problema e a capacidade do estudante. Por exemplo, uma funcionalidade de geração de exercícios baseados em parâmetros definidos pelo professor. Segundo, relacionado à interface, ter mais modos de interação disponíveis para melhorar tarefas de colaboração para monitoramento do progresso da performance do aluno.

## Referências

- Baker, F. B. (2001). *The Basics of Item Response Theory*. ERIC Clearinghouse on Assessment and Evaluation, University of Wisconsin.
- Baker, R. S., Goldstein, A. B., and Heffernan, N. T. (2010). Detecting the Moment of Learning. *LNCS*, 6094(PART I):25–34. Springer-Verlag Berlin Heidelberg.
- Champaign, J. and Cohen, R. (2010). A Model for Content Sequencing in Intelligent Tutoring Systems Based on the Ecological Approach and Its Validation Through Si-

- mulated Students. pages 486–491. Association for the Advancement of Artificial Intelligence (AAAI).
- Corbett, A. T. and Anderson, J. R. (1994). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4):253–278.
- Direne, A. (1990). Authoring Intelligent Systems for Teaching Visual Concepts. *International Journal of Artificial Intelligence in Education*, 1(4):3–14.
- Isotani, S., Adams, D., Mayer, R. E., Durkin, K., Rittle-Hohnson, B., and McLaren, B. M. (2011). Can Erroneous Examples Help Middle-School Students Learn Decimals? volume 6964 of *Lecture Notes in Computer Science*, pages 181–195, Palermo. Springer Berlin Heidelberg.
- Major, N. and Reichgelt, H. (1992). COCA: A Shell for Intelligent Tutoring Systems. In *Proc. of the International Conference on Intelligent Tutoring Systems (ITS92)*, pages 523–530. Springer.
- Marczal, D. (2014). Farma: Uma ferramenta de autoria para objetos de aprendizagem de conceitos matemáticos. Tese de doutorado, Universidade Federal do Paraná, UFPR.
- Marczal, D. and Direne, A. (2012). FARMA: Uma Ferramenta de Autoria para Objetos de Aprendizagem de Conceitos Matemáticos. In *Anais do Simpósio Brasileiro de Informática na Educação*, volume 23.
- Mitrovic, A. (2003). An intelligent SQL tutor on the web. *International Journal of Artificial Intelligence in Education*, 13(3):173–197.
- Pinkwart, N. and Loll, F. (2009). Comparing three approaches to assess the quality of students' solutions. In Dicheva, D., Mizoguchi, R., and Pinkwart, N., editors, *AIED 2009 Workshops Proceedings Volume 2, SWEL'09: Ontologies and Social Semantic Web for Intelligent Educational Systems Intelligent Educational Games*, pages 81–85.
- Ravi, G. A. and Sosnovsky, S. (2013). Exercise difficulty Calibration Based on Student Log Mining. In Mödritscher, F., Luengo, V., Law, E. L.-C., and Hoppe, U., editors, *Proceedings of DAILE'13: Workshop on Data Analysis and Interpretation for Learning Environments*, Villard-de-Lans (France).
- Ritter, S., Anderson, J. R., Koedinger, K. R., and Corbett, A. (2007). Cognitive tutor: Applied research in mathematics education. *Psychonomic bulletin & review*, 14(2):249–255.
- Schatten, C. and Schmidt-Thieme, L. (2014). Adaptive Content Sequencing without Domain Information. *6th International Conference on Computer based Education*.
- Segal, A., Katzir, Z., Gal, K., Shani, G., and Shapira, B. (2014). EduRank: A Collaborative Filtering Approach to Personalization in E-learning. In Stamper, J., P. Z. M. M. M. B., editor, *Proceedings of the 7th International Conference on Educational Data Mining*, pages 68–75.
- Self, J. (1990). Bypassing the Intractable Problem of Student Modelling. (41):1–26. AAI/AI-ED.