UM FRAMEWORK PARA A GERAÇÃO DE REPERTÓRIOS DE ENSINO INDIVIDUALIZADO BASEADO EM DIFICULDADE ADAPTATIVA

Gilberto Nerino de Souza Junior
Programa de Pós-Graduação de Engenharia Elétrica - Universidade Federal do Pará
Rua Augusto Corrêa, 1 - Guamá, Belém - PA, 66075-110
gilbertonerinojr@gmail.com

Pedro Paulo Furtado Oliveira Junior
Instituto de Tecnologia- Universidade Federal do Pará
Rua Augusto Corrêa, 1 - Guamá, Belém - PA, 66075-110
p.paulo.f.oliveira@gmail.com

Dionne Cavalcante Monteiro
Instituto de Ciências Exatas e Naturais- Universidade Federal do Pará
Rua Augusto Corrêa, 1 - Guamá, Belém - PA, 66075-110
dionne@ufpa.br

Ádamo Lima de Santana
Programa de Pós-Graduação de Engenharia Elétrica - Universidade Federal do Pará
Rua Augusto Corrêa, 1 - Guamá, Belém - PA, 66075-110
adamo@ufpa.br

RESUMO

Em sistemas computacionais para o ensino individualizado geralmente um especialista educacional necessita criar manualmente um repertório de tarefas baseado na sua experiência e observando as habilidades e limitações de aprendizado dos alunos. Para um repertório de ensino adequado também é importante identificar a dificuldade das tarefas de ensino que melhor se adaptem ao conhecimento do aluno. Este artigo apresenta um framework para a geração automática de repertórios de tarefas de ensino de forma individualizada baseado em dificuldade adaptativa e no programa de ensino ALEPP (Aprendendo a Ler e Escrever em Pequenos Passos). Foram realizados experimentos com o intuito de identificar a estabilidade de um algoritmo genético proposto para a criação de tarefas. Resultados mostraram que o framework é capaz de gerar tarefas com dificuldade condizente ao nível de desempenho de alunos.

PALAVRAS CHAVE. Repertório de Ensino, Algoritmo Genético, Dificuldade Adaptativa

Tópicos : EDU - PO na Educação, OA - Outras aplicações em PO.

ABSTRACT

In computer systems for the individualized teaching an educational expert need to create manually a repertoire of tasks based on his experience and observing the skills and limitations of learning student. To generate a teaching repertoire proper it is important to identify the difficulty of teaching tasks best suited to the student's knowledge. This article presents a framework for automatic generation of repertoires of teaching tasks based on adaptive difficulty and the teaching program ALEPP (Learning to Read and Write in Small Steps).Experiments were performed in order to identify the stability of the genetic algorithm proposed. Results showed that the framework is capable of generating tasks with difficulty in keeping the level of student performance.

KEYWORDS. Teaching Repertoire, Genetic Algorithm, Adaptive Difficulty

Paper topics : EDU - OR in education, OA - Another applications on OR

1. Introdução

Ao longo das duas últimas décadas observou-se um avanço tecnológico dos sistemas de aprendizado e tutoria inteligente [Kumar e Kim 2014]. Apesar deste progresso e da crescente utilização de tais sistemas, segundo o último relatório da UNESCO (*United Nations Educational, Scientific and Cultural Organization*), aproximadamente 250 milhões de 650 milhões de crianças em idade escolar primária não sabem ler [United Nations 2014].

Alguns pesquisadores sugerem que a criação de tarefas de ensino para alunos com déficit de aprendizado deve seguir um ensino individualizado [Skinner 1950], [Silva 2012]. Vários programas computacionais utilizam esta abordagem individualizada e buscam auxiliar pessoas com alguma limitação de aprendizado [Orlando 2009], [Sampayo et al. 2013], [Ronimus et al. 2014].

Estes programas utilizam práticas cognitivas e comportamentais para a criação de tarefas de ensino geralmente construídas de maneira manual. Ou seja, um especialista educacional deve criar um conjunto de tarefas baseado na experiência de cada aluno, observando as habilidades e limitações de aprendizado.

Este conjunto de tarefas compõem um repertório de ensino individualizado, mas para isto, é necessário identificar as tarefas de ensino que melhor se adaptem ao aluno [Marković e Jovanović 2012], [Millán, et al. 2013], [Feldman, et al. 2014]. Logo, determinar os atributos de "dificuldade" dos itens de ensino é um dos componentes de maior relevância para o ensino de um conteúdo personalizado e factível ao aluno [Conejo et al. 2014].

Este artigo tem como proposta apresentar um framework para a geração automática de repertórios de ensino individualizado, baseado em dificuldade adaptativa. O framework tem como base a metodologia de ensino do ALEPP (Aprendendo a Ler e Escrever em Pequenos Passos) [Rose et al. 1989] e busca, através de um algoritmo genético (AG), gerar um conjunto de tarefas adaptadas baseada na configuração prévia inserida pelo especialista de ensino.

2. O programa de ensino - ALEPP

O programa de ensino "Aprendendo a Ler e a Escrever em Pequenos Passos" (ALEPP) [Rose et al. 1989] busca auxiliar pessoas que possuem dificuldade em leitura. O ALEPP estabelece uma relação entre palavras ditadas, imagens, palavras impressas e respostas através de tarefas de emparelhamento ao modelo de referência (MTS - *Matching-To-Sample*) [Mackay 1985]. Em tarefas MTS, quando há a presença de um estímulo modelo, o aluno deve selecionar uma alternativa correspondente ao modelo [Souza et al. 2009].

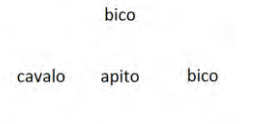
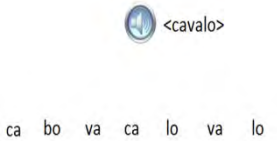
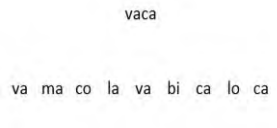
No ALEPP também são utilizados as tarefas CRMTS (*Constructed Response* - MTS) [Siqueira 2012] onde se apresentam estímulos modelos e diferentes estímulos de comparações. Este procedimento é muito utilizado para montar palavras sílaba por sílaba até apresentar a palavra totalmente construída.

Um conjunto de tarefas podem ser agrupadas para formar um repertório de ensino a fim de promover o aprendizado. Contudo o programa de ensino não possui uma forma automática de gerar repertórios de ensino com dificuldade adaptativa, limitando sua aplicação em salas de aulas que possuem muitos alunos. Este trabalho busca preencher esta lacuna para auxiliar o especialista educacional a montar o seu repertório de ensino. Na Tabela 1 são exibidos exemplos de tarefas de ensino utilizadas neste trabalho.

3. Trabalhos relacionados

Algumas pesquisas relacionadas ao tema, geração de repertório de ensino com dificuldade adaptativa, podem ser identificadas e categorizadas em duas classes: os sistemas de ensino adaptáveis e os sistemas de ensino adaptativos. Nos sistemas de ensino adaptáveis o tutor necessita customizar a sessão de ensino de maneira manual para o estudante [Marković e Jovanović 2012].

Tabela 1. Exemplos dos tipos de tarefa de ensino do procedimento MTS e CRMTS

Ilustração da tarefa	Tipo da tarefa e Função do Aluno	Ilustração da tarefa	Tipo da tarefa e Função do Aluno
	Tipo de tarefa: AB. Diante de uma instrução, o aluno deverá selecionar a figura do «bolo».		Tipo de tarefa: CB. Diante da palavra escrita «cavalo», o aluno deverá selecionar a figura «cavalo».
	Tipo de tarefa: AC. Diante de uma instrução, o aluno deverá selecionar a palavra escrita «tatu».		Tipo de tarefa: BB. Diante da figura «mala», o aluno deverá selecionar a figura «mala».
	Tipo de tarefa: BC. Diante da figura de um apito, o aluno deverá selecionar a palavra escrita «apito».		Tipo de tarefa: CC. Diante da palavra escrita «bico», o aluno deverá selecionar a palavra «bico».
	Tipo de tarefa: AE. Diante de uma instrução, o aluno deverá compor a palavra «cavalo», escolhendo as sílabas na ordem correta.		Tipo de tarefa: BE. Diante de uma instrução, o aluno deverá compor a palavra «tomate», escolhendo as sílabas na ordem correta.
	Tipo de tarefa: CE. Diante de uma instrução, o aluno deverá compor a palavra «vaca», escolhendo as sílabas na ordem correta.		

Estes sistemas exigem que um tutor esteja presente ao lado do aluno de maneira integral ou semi-integral. O tutor observa o comportamento e habilidades do aluno para propor melhores tarefas baseado no seu conhecimento pessoal.

O aplicativo GEIC (Gerenciador de Ensino Individualizado por Computador) integra um sistema que permite a alteração do fluxo de ensino a partir de regras pré-estabelecidas [Orlando 2009]. O aplicativo já foi empregado em escolas para alunos com ou sem problemas na aquisição da leitura. Pode-se considerar que o GEIC informatizou o processo do programa de ensino ALEPP [Souza et al. 2009].

O artigo de Sampayo et al. [2013] aborda a relação entre motivação e ajuste da dificuldade em um jogo computacional comparado a um texto impresso. Os resultados sugerem que os estudantes, que passaram pelo jogo com ajuste de dificuldade manual, tiveram melhores taxas de aprendizado. O autor explica que um ajuste na dificuldade pode promover o aprendizado de leitura e escrita.

A intervenção manual para a construção de bloco de tarefas pode ser considerada uma limitação, pois o instrutor necessita construir um determinado conjunto de tarefas para cada aluno levando em conta seu desempenho e suas habilidades. O especialista (professor, instrutor ou psicólogo) é indispensável para o ensino, mas, naturalmente pode tornar os processos de modelagem dos repertórios de ensino mais demorados e custosos [Cruz e Both 2012], [Pedri 2012].

Já os sistemas de ensino adaptativos são focados na adaptação do aprendizado baseado na observação do conhecimento do aluno [Marković e Jovanović 2012]. Com isto, o perfil do aluno não é considerado estático, mas é constantemente acompanhado.

Pode-se citar os trabalhos de Chen [2008] e Chen et al. [2008], que mostram um sistema para o ensino de frações matemáticas, programação e inteligência artificial. Neste sistema, são recomendados caminhos personalizados de acordo com as habilidades do aluno e complexidade do curso. O sistema usa algoritmos genéticos para estabelecer um caminho de aprendizado baseado em respostas incorretas de estudantes. Experimentos mostraram que a aplicação proposta foi superior ao modo de aprendizado livre, ou seja, quando o aluno escolhe os cursos que deseja aprender.

O trabalho de Verdú et al. [2012] buscou uma abordagem voltada para ambientes competitivos de aprendizagem. Em um domínio específico de competição, o ajuste de dificuldade é feito de forma empírica com auxílio do especialista como fonte primária. O artigo usou um sistema fuzzy aliado a um algoritmo genético para classificar a dificuldade das questões como: fácil, moderado e difícil. Contudo a pesquisa não executou a identificação prévia do nível de conhecimento do aluno.

A pesquisa de Souza et al. [2012] buscou classificar o aprendizado com o mapeamento da dificuldade no ensino da leitura através de tarefas de emparelhamento ao modelo. O grau de conhecimento de um aluno em uma determinada palavra e o seu aprendizado foi obtido através da técnica de regressão logística. Esta técnica foi posteriormente reformulada e usada no trabalho de Pereira et al. [2013], onde buscou gerar uma tarefa para o aluno utilizando lógica fuzzy. A aplicabilidade do sistema possui uma lacuna no fato de não abordar possíveis regras para a criação de um repertório completo de tarefas.

Outro estudo interessante é o de Lin et al. [2013], que resultou em um caminho de aprendizagem personalizado baseado em árvore de decisão. O resultado mostrou que uma árvore de decisão, definida através do algoritmo C4.5, melhorou as taxas de acerto dos estudantes universitários. No entanto, o sistema é focado em pessoas que já sabem ler (adultos) onde a dificuldade das tarefas de ensino não foram abordadas.

É importante destacar que poucos dos sistemas inteligentes propostos são focados para crianças, seja com ou sem problemas de aprendizado na aquisição da leitura. Nenhum dos trabalhos citados utilizou quesitos da análise comportamental, mas alguns utilizaram modelos de inteligência computacional como proposta de otimizar algum fator de aprendizagem.

4. Dificuldade Dinâmica da tarefa

A dificuldade dinâmica da tarefa de ensino é um valor adaptativo alterado pelo desempenho do aluno, extraídos de seus dados históricos. Na presente proposta, a dificuldade dinâmica de uma tarefa para um determinado aluno aumenta a medida que a probabilidade de erro da tarefa aumenta. Logo, é proposto a Equação (1) que representa a dificuldade dinâmica de uma tarefa de ensino levando em consideração as habilidades do aluno e a dificuldade estática.

$$Dd_j^i = De^i(1 - P_j^i) \tag{1}$$

Onde: Dd_j^i é a dificuldade dinâmica da tarefa i de um aluno j; De^i é a dificuldade estática da tarefa i; P_j^i é a probabilidade de sucesso/acerto da tarefa i para um aluno j. Nas próximas subseções serão explanadas os termos referentes a probabilidade de erro e dificuldade estática.

4.1. Dificuldade Estática da tarefa

Pode-se definir dificuldade estática como um valor constante e de mesmo peso para dois indivíduos em fases diferentes do processo de aprendizado. A dificuldade estática de uma tarefa é composta por 3 fatores de dificuldade identificados e modelados em conjunto com especialistas do domínio de aplicação, Equação (2).

$$De^i = Dt^i + Dn^i + Ds^i \tag{2}$$

Onde: Dt^i é o fator de dificuldade relacionado ao tipo de tarefa da tarefa i; Dn^i é o fator de dificuldade referente ao número de escolhas/comparações da tarefa i; Ds^i é o fator de dificuldade referente a similaridade entre as palavras da tarefa i.

O primeiro fator de dificuldade da tarefa, Dt^i, leva em consideração o peso estipulado da tarefa i que é calculada pela Equação(3).

$$Dt^i = \frac{wt^i - wt_{menor}}{wt_{maior} - wt_{menor}} \tag{3}$$

Onde: wt^i é o peso do tipo de tarefa para a tarefa i; wt_{maior} é o maior peso dos tipos de tarefas; wt_{menor} é o menor peso dos tipos de tarefas. Esse fator de dificuldade leva em conta a relação onde cada peso do tipo de tarefa equivale a dificuldade de um determinado tipo de tarefa. A soma desses pesos é igual a 1.

O segundo fator de dificuldade, Dn^i, faz referência ao número de escolhas da tarefa i e é representado pela Equação (4). A ideia dessa equação leva em consideração que à medida que se aumenta o número de possíveis escolhas aumenta-se a dificuldade de uma tarefa. É importante ressaltar que para tarefas MTS o número de comparações (n) geralmente não excede 5 escolhas e para tarefas CRMTS raramente as escolhas passam de 15 opções para testes comportamentais.

$$Dn^i = 1 - \frac{1}{n_e{}^i + \delta_n}, \quad \begin{cases} 15 > n > 0 \\ 1 > \delta_n > 0 \end{cases} \tag{4}$$

Onde: δ_n é uma taxa de crescimento de dificuldade para o componente Dn^i(assumiu-se $\delta_n = 0{,}1$ adquirido empiricamente); $n_e{}^i$ é o número de escolhas de comparações da tarefa i.

O terceiro fator de dificuldade a ser apresentado é o Ds^i, Equação (5), que refere-se à similaridade entre palavras e é calculada ao se comparar o estímulo modelo com os outros estímulos da tarefa. Este fator de dificuldade está ligado a dificuldade em diferenciar estímulos parecidos devido ao fato que o aluno pode confundir a opção correta caso existam outras opções similares de escolha. Portanto, quanto maior a similaridade de um estímulo de comparação em relação ao estímulo modelo, maior será a dificuldade de escolha da opção correta.

$$Ds^i = \frac{\sum_{p=1}^{n_s} s_p}{n_s} \tag{5}$$

Onde: s_p é a similaridade da escolha modelo com a escolha de comparação p; n_s é o número de similaridades calculadas.

Este fator de dificuldade é expresso de formas diferentes conforme o procedimento MTS ou CRMTS. No procedimento MTS um valor de similaridade é calculado tendo como entrada o estímulo modelo e o estímulo de comparação.

Para exemplificar utiliza-se a tarefa do exemplo exibido na Figura 1(a). Nesta tarefa a palavra estímulo modelo é "bolo" em formato de imagem e as palavras de comparações são a palavra "bola" e palavra "tatu" em formato textual. O aluno deve realizar a comparação entre a palavra modelo "bolo" com as palavras "bola" e "tatu". Subtende-se que o aluno realiza no mínimo 2 comparações entre os estímulos. Após as comparações o aluno escolhe qual palavra é referente ao modelo.

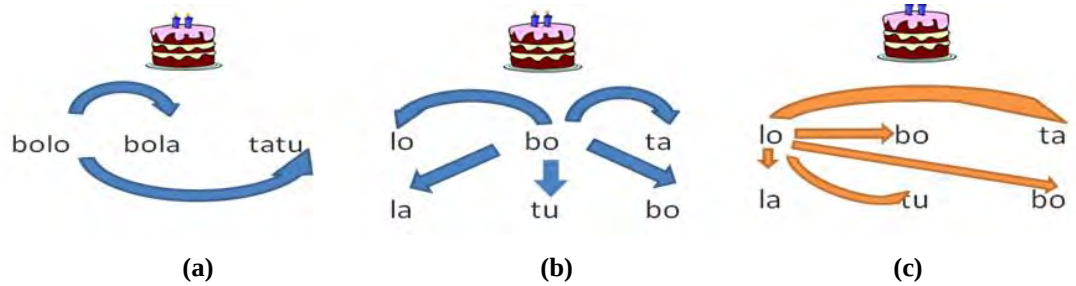


Figura 1. A figura representa um esquema de coleta das similaridades de tarefas do procedimento MTS e CRMTS. As setas representam a ação de comparação para calcular a similaridade desta tarefa. (a) similaridade para MTS (b)similaridades para a primeira sílaba "bo" para o procedimento CRMTS, (c) similaridades da segunda sílaba "lo" para o procedimento CRMTS.

No exemplo da Figura 1(a), a similaridade da palavra "bolo" com a palavra "bola" é 0.8 e da palavra "bolo" com a palavra "tatu" é 0.1. Esse valor é chamado de similaridade entre estímulos de palavras ou simplesmente de similaridade de palavras. Valores iguais a 1 representam 100% de similaridade, ou seja, estímulos exatamente iguais. Para este exemplo, o resultado é igual a 0.45, conforme resolução da Tabela 2.

Tabela 2. Exemplo do cálculo do fator de similaridade da tarefa da Figura 1(a).

Número de Similaridades	Primeiro Estímulo	Segundo Estímulo	Similaridade	Fator de similaridade da tarefa da Figura 1(a)
2	"bolo"	"bola"	0.8	$Ds = \frac{(0.8 + 0.1)}{2} = 0.45$
	"bolo"	"tatu"	0.1	

No procedimento CRMTS os estímulos para escolhas são sílabas. Para cada sílaba do modelo é extraído a similaridade em relação as outras sílabas. A Figura 1(b) e Figura 1(c) representa a coleta de similaridades para as sílabas da palavra "bolo", Tabela 3. Percebe-se a semelhança entre a coleta de similaridades de tarefas com procedimento MTS. Neste caso é necessário calcular a similaridade entre todas as sílabas do modelo em relação as outras sílabas de escolhas. Quando os estímulos são 100% iguais, converte-se a similaridade igual de1 para 0, visto que isto não agrega dificuldade, mais sim facilidade para o aluno que possui mais opções corretas como escolhas.

Tabela 3. Exemplo do cálculo do fator de similaridade da tarefa da Figura 1 (b) e Figura 1 (c).

Número de Similaridades	Primeiro Estímulo	Segundo Estímulo	Similaridade	Fator de similaridade da tarefa da Figura 1 (b) e Figura 1 (c)
10	"bo"	"ta"	0.1	$Ds = \frac{(0.1 + 0.5 + 0.1 + 0.2 + 0) + (0.6 + 0.1 + 0.6 + 0.2 + 0.2)}{5 + 5} = 0.26$
	"bo"	"lo"	0.5	
	"bo"	"la"	0.1	
	"bo"	"tu"	0.2	
	"bo"	"bo"	1 (converte-se para 0)*	
	"lo"	"bo"	0.6	
	"lo"	"ta"	0.1	
	"lo"	"bo"	0.6	
	"lo"	"tu"	0.2	
	"lo"	"la"	0.2	

A similaridade pode ser calculada por funções de similaridade e proximidade entre palavras [Levenshtein 1965] ou por uma tabela com as similaridades de palavras sugeridas por especialistas[Souza et al. 2012].

4.2. Probabilidade de Acerto da tarefa

Para encontrar a probabilidade de acerto de uma tarefa este trabalho propõe a utilização de um modelo inspirado no TRI (Teoria de Resposta ao Item) [Sharkness 2014], que leva em consideração os fatores de dificuldade estática de uma tarefa e o conhecimento do aluno. O modelo é exibido na Equação (6).

$$P_j^i = \frac{1}{1 + e^{-1,7(\theta - De^i)}} \tag{6}$$

Onde: De^i é a dificuldade estática da tarefa i ; θ é o conhecimento do aluno j.

A variável θ representa o conhecimento do aluno em realizar determinadas tarefas de teste. Este valor é determinado baseado nos dados de respostas históricas do aluno, como por exemplo, o resultado ponderado da execução de um pré-teste de tarefas. Para a utilização da fórmula da Equação (6), a dificuldade estática e o conhecimento do aluno podem ser normalizados em um intervalo entre -3 a 3.

5. Framework para Geração de Repertório

Em geral, a geração de repertórios de ensino pode ser feita através de dois métodos: empírico e por força bruta. O método empírico é subjetivo e depende de um especialista para gerar cada tarefa encadeando-as até formar uma lista de tarefas baseado no seu conhecimento tácito e na identificação do conhecimento do aluno. Este processo pode ser custoso para um especialista educacional, que necessita implementar o ensino para diferentes alunos. Outra forma seria utilizar a força bruta gerando todas as possíveis tarefas e formando os possíveis repertórios dentro de um escopo de conteúdo de ensino, restrições e conhecimento do aluno. Com o método por força bruta seria necessário testar todas as possíveis combinações das tarefas. Tendo como referência o programa de ensino ALEPP, para formar um conjunto de tarefas possíveis seria necessário combinar atributos como: Palavras Modelo, Número de palavras de comparações e Tipos de tarefas. Realizando a combinação destes atributos para uma única tarefa temos a seguinte Equação(7):

$$T_C = \left(\sum_{Nc_{min}}^{Nc_{max}} \left(\frac{N_{max}\,!}{Nc_{min}!(Nc_{max} - Nc_{min})!}\right)\right) * T_{max} * M_{max} \qquad (7)$$

Onde: T_C é o número total de combinações de tarefas possíveis; N_{max} é o número máximo palavras utilizadas na sessão de ensino; Nc_{max} éo número máximo de comparações de uma tarefa; Nc_{min} é o número mínimo de comparações de uma tarefa; T_{max} é o número máximo de tipos de tarefas; M_{max}é o número máximo de palavras modelos utilizadas no ALEPP.

Para o ALEPP, sendo executado com N_{max} = 96, T_{max}= 9, M_{max} = 96 e Nc_{max} = 5,o resultado dessa combinação seria aproximadamente $T_C = 5.58^{10}$combinações de tarefas. O tempo para execução das combinações tornaria inviável esta abordagem para um computador comum.

Baseado nas considerações descritas, identificou-se que o uso de meta-heurísticas permitiria a busca de uma solução viável para o problema de geração de repertórios de ensino. Para tal foi escolhido o Algoritmo Genético, pela sua praticidade em tratar de dados discretos e a possibilidade para convergir para um ponto aceitável [Rothlauf 2006].

5.1. Modelagem do Framework

O Framework de geração de tarefas é representado na Figura 2. A imagem mostra as atividades de geração de tarefas.

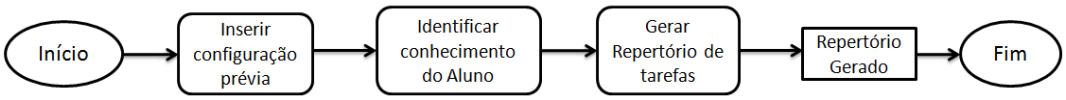


Figura 2. Fluxo de geração de um repertório de tarefas de ensino.

O processo começa com a inserção dos dados prévios para o ensino. Para isso, o especialista deve preencher todos os dados necessários para a execução do sistema, que são:

- Palavras Modelos: uma lista de palavras modelos que se deseja ensinar;
- Tipos de tarefas: uma lista com os tipos de tarefas que formarão o repertório gerado;
- Palavras de comparação: uma lista com todas as palavras como alternativas para a comparação com o modelo;
- Pré-Teste: um conjunto de tarefas que o aluno já executou;
- Fluxo de dificuldade: um vetor com a curva de dificuldade do repertório de ensino, onde cada posição armazena um valor de dificuldade das tarefas de ensino a serem geradas.

5.2. Fluxo de Dificuldade

O fluxo de dificuldade escolhido pelo especialista representa um vetor sequencial de dificuldade das tarefas que desejasse gerar. Esta estrutura é um vetor F com valores que representam a dificuldade da tarefa conforme a Equação (8).

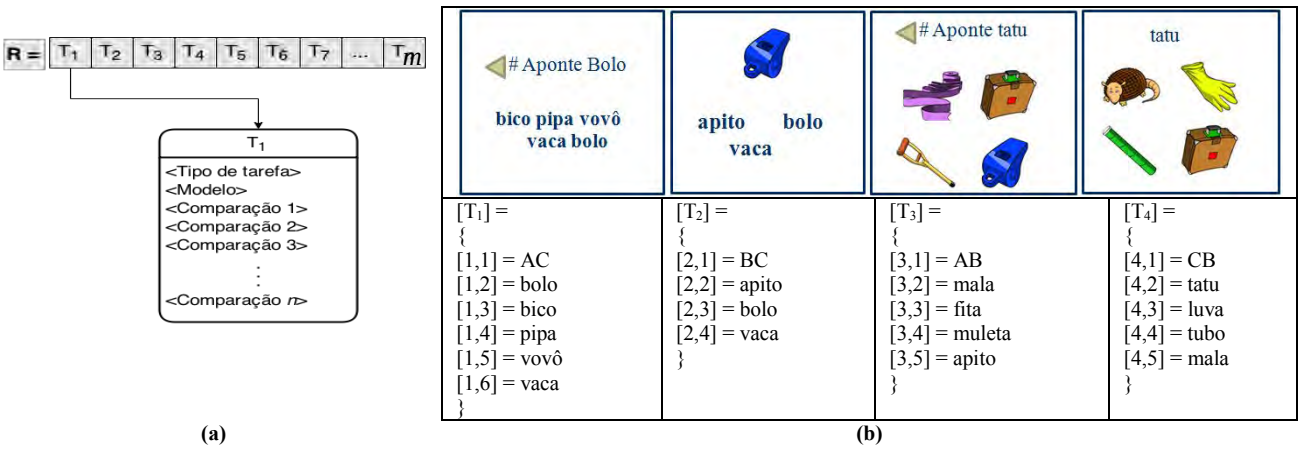
$$\mathbf{F} = [\, F_1, F_2, F_3, F_4, F_5, F_6, F_7, \ldots, F_m \,] \tag{8}$$

O fluxo de dificuldade é a curva de dificuldade das tarefas de ensino que serão geradas e a dificuldade do conjunto de tarefas que o especialista deseja apresentar ao aluno. Dessa forma, o especialista pode gerar um repertório com *m* tarefas. Pode-se utilizar diversas variações de fluxos de dificuldade personalizadas, como por exemplo: o fluxo sigmoidal, o fluxo crescente linear, o fluxo degrau e o fluxo senoidal.

A busca de um repertório com dificuldade próxima ao fluxo de dificuldade é o alvo do módulo de geração de tarefas com implicação na realização da busca das tarefas que melhor se adaptem ao vetor F, sem desconsiderar as habilidades do aluno e as restrições do problema.

5.3. Estrutura do Indivíduo

Cada repertório de tarefas de ensino é modelado em formato de lista, onde T representa a tarefa do repertório e **R** é a codificação de um repertório concebido pela estrutura de dados, Figura 3(a). Para fim de ilustração, um exemplo de repertório é apresentado na Figura 3(b) que mostra um repertório de 4 tarefas e sua codificação no formato de uma lista de células.



(a) **(b)**

Figura 3.(a)codificação de um repertório de tarefas de ensino, (b)exemplo de codificação de um repertório de ensino.

5.3. Função de *Fitness*

O objetivo para a solução do problema é encontrar um repertório onde a dificuldade dinâmica das tarefas estão próximas ao fluxo de dificuldade pré-estabelecido. Portanto é necessário encontrar a menor distância entre o fluxo de dificuldade e a dificuldade dinâmica das tarefas. A função de *fitness* é expressa na Equação(9).

$$\mathbf{F_F}(\mathbf{R}, \mathbf{F}) = \boldsymbol{\rho m} - \left(\sum_{\mathbf{i=1}}^{\boldsymbol{m}} (|\boldsymbol{Dd}(\mathbf{R_i}) - \mathbf{F_i}|) + \sum_{\mathbf{j=1}}^{\boldsymbol{\rho}} \Phi_{\mathbf{j}} \right) \tag{9}$$

Onde: F_F é o valor de *fitness* do indivíduo; $\mathbf{R}_i$ é a *i*-ésima tarefa do repertório **R**; $\mathbf{F}_i$ é o *i*-ésimo valor do fluxo de dificuldade **F**; $Dd(R_i)$ é o valor de dificuldade dinâmica da *i*-ésima tarefa; Φ_j refere-se penalidade da *j*-ésima restrição do problema; *m* é número de tarefas do repertório a ser gerado; ρ é número de penalidades.

Para obedecer às restrições do problema, foi proposto a aplicação de penalidades no *fitness*. As penalidades Φ_1, Φ_2, Φ_3 e Φ_4 buscam dar ao repertório um balanceamento com diversidade de tarefas visando diminuir a importância de repertórios com tarefas ou atributos repetidos, exibidos na Tabela 4.

Tabela 4. Penalidades da função de *fitness*

Identificador	Descrição
Φ_1	Penalidade por repetições de número de tarefas. Penaliza-se o repertório/ indivíduo com o número que representa a soma de tarefas iguais do repertório.
Φ_2	Penalidade por repetições de modelos. Penaliza-se o repertório/ indivíduo com o número que representa a soma de tarefas com modelos iguais.
Φ_3	Penalidade por repetições de tipos de tarefas. Penaliza-se o repertório/ indivíduo com o número que representa a soma de tarefas com tipos de tarefas iguais.
Φ_4	Penalidade por repetições de modelos e tipos de tarefas. Penaliza-se o repertório/ indivíduo com o número que representa a soma de tarefas com tipos de tarefas e modelos iguais.

6. Experimentos e Resultados

Experimentos simulados foram propostos para mostrar a confiabilidade e estabilidade do AG. O experimento consiste em uma análise baseado na configuração prévia de conteúdo de ensino apresentada na Tabela 5. Foi levado em consideração 3 pré-testes que representam: Baixo desempenho, com alunos que obtiveram baixa taxa de acertos; Médio desempenho, com alunos que obtiveram taxas médias de acertos; Alto desempenho, com alunos que obtiveram taxas alta de acertos.

Tabela 5. Parâmetros prévios de configuração do conteúdo de ensino para a geração de tarefas.

Item	Descrição / Valor
Palavras de comparação	[bolo tatu vaca bico mala tubo pipa luva fita pato]
Palavras Modelos	[mala tatu]
Tipos de Tarefas	[AB AC BC CB]
Número de Tarefas	27
Número mínimo de palavras como escolhas	2
Número máximo de palavras como escolhas	4

Em relação a configuração do AG, as operações de seleção foram: a seleção torneio e o elitismo. O método da operação de cruzamento foi o esquema uniforme onde há troca de cada gene conforme Rothlauf [2006]. No método de mutação proposto, cada atributo do gene é escolhido aleatoriamente e substituído por um outro parâmetro do atributo, também escolhido aleatoriamente. As taxas de mutação e cruzamento são auto adaptadas conforme a diversidade da população [Zhu, 2003]. O algoritmo foi executado com 200 iterações e 200 indivíduos na população.

A Figura 4 apresenta a convergência para 100 execuções do AG e a dificuldade adaptada dos repertórios gerados de uma execução para cada pré-teste. Ao analisar os gráficos evolutivos do algoritmo genético é possível identificar a convergência com pouca variação de *fitness* dos melhores indivíduos. Os gráficos evolutivos obtiveram comportamento similares mesmo para diferentes desempenhos de aprendizado estipulados pelos pré-testes.

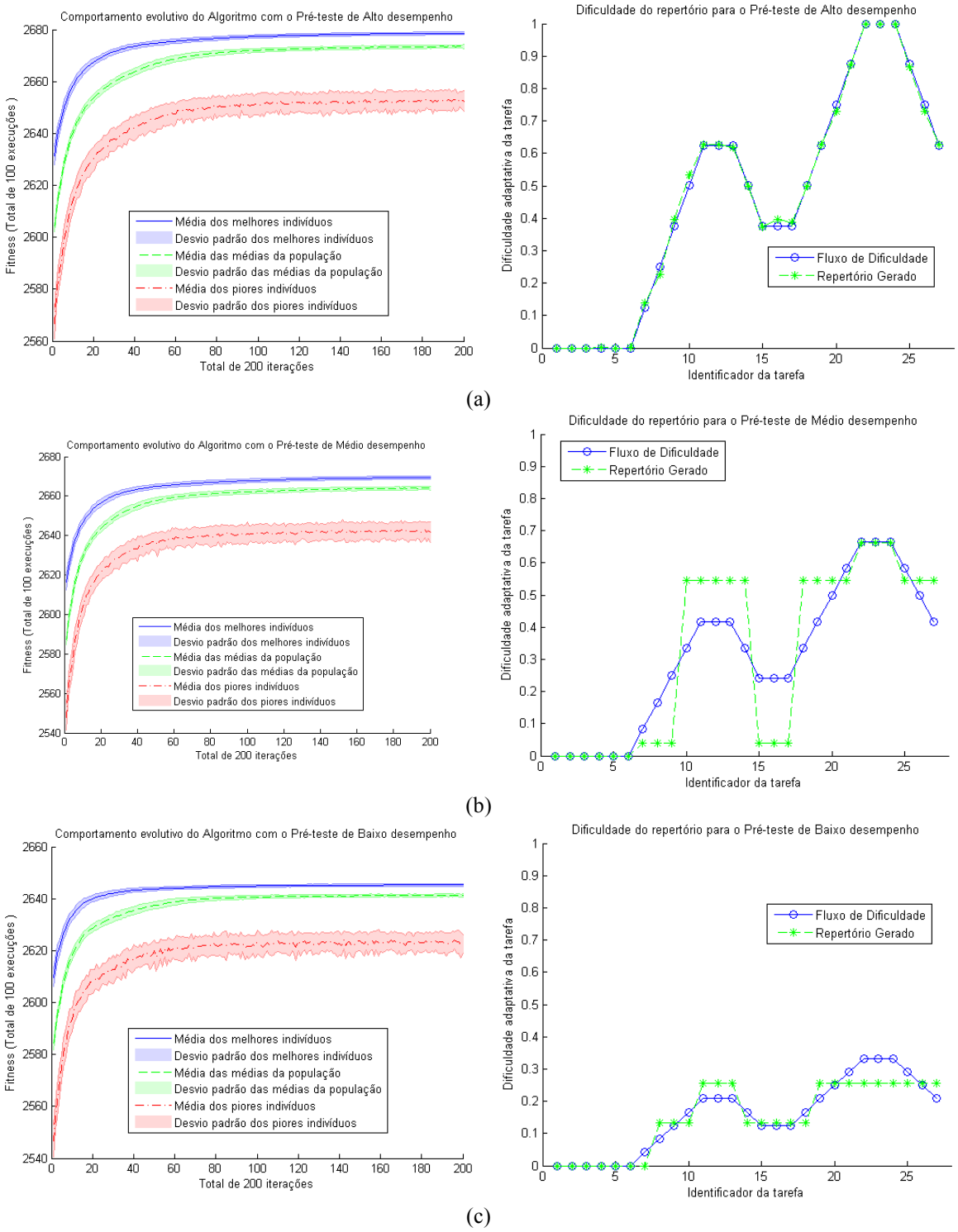


Figura 4. Os gráficos mostram a convergência do AG (primeira coluna) e comportamento da curva de dificuldade dos repertórios gerados para os três pré-testes de **(a)** Baixo Desempenho, **(b)** Médio Desempenho e**(c)** Alto Desempenho.

Para o pré-teste de Alto desempenho, observa-se uma distância bem pequena entre o repertório gerado em relação ao fluxo de dificuldade sugerido. Isto tende a ser uma constante pois o algoritmo realiza a busca de tarefas com diversos tipos de tarefas e palavras de comparações diversificados visto que o aluno possui alto desempenho e tem capacidade de realizar tarefas variadas e com dificuldades maiores.

Já para o pré-teste de médio e baixo desempenho, observou-se uma distância pequena entre os repertórios gerados e o fluxo de dificuldade, mas não tão próxima quando comparado ao pré-teste de alto desempenho. Nos dois casos, o algoritmo gera tarefas que seguem o fluxo de dificuldade mesmo não encontrando tarefas realmente próximas. Isto ocorre devido a não existência de tarefas consideradas muito complexas para alunos com baixo ou médio desempenho.

7. Conclusão

Esta pesquisa apresentou um framework capaz de realizar a geração de tarefas de ensino baseado em dificuldade adaptativa e na metodologia de ensino do ALEPP. Um algoritmo genético foi desenvolvido e compõem o núcleo do framework que, baseado em configurações prévias inseridas por um especialista de ensino, permitiu realizar uma análise das soluções geradas.

As dificuldades das tarefas são adaptativas e dentro do escopo de configuração de conteúdo de ensino do especialista, obedecendo à estrutura base de tarefas do procedimento MTS. Resultados indicam que o algoritmo genético modelado se manteve estável gerando poucas variações de repertórios no número total de execuções e com convergência satisfatória. As dificuldades dos repertórios também se mantiveram de acordo com o nível de aprendizado dos alunos identificado pelo pré-teste.

O framework busca atender o ensino individualizado, porém sugestões para novas pesquisas podem ser consideradas, como por exemplo, a otimização da qualidade dos parâmetros de dificuldade, além da aplicação prática dos repertórios gerados em ambientes reais de aprendizado. Por fim, com a construção do framework proposto, este estudo pretende auxiliar na diminuição de crianças com déficit de aprendizado da leitura e servir de apoio para professores, psicólogos e tutores de ensino, podendo ser aplicado em outras áreas de ensino com algumas adaptações.

8. Agradecimentos

Este trabalho foi realizado com apoio Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) com auxílio de bolsa de estudo. Todo o conteúdo apresentado neste material é de inteira responsabilidade dos autores.

Referências

Chen, C. M. (2008). Intelligent web-based learning system with personalized learning path guidance. *Computer and Education*, 51:787–814.

Chen, C. M. e Duh, L. J. (2008). Personalizedweb-based tutoring system based on fuzzy item response theory. *Expert Systems with Applications*, 34:2298–2315.

Conejo, R., Guzmám, E., Perez, J. L. C. e Barros, B. (2014). An empirical study on the quantitative notion of task difficulty. *Expert Systems with Applications*, 41:594–606.

Cruz, R. C. N. e Both, I. J. (2012). A importância do tutor em Educação a Distância (EaD). *Caderno Intersaberes*, 32:5–44.

Feldman, J., Monteserin, A. e Amandi, A.. (2014). Automatic detection of learning styles: state of the art. *Artificial Intelligence Review*, 4 :157–186.

Kumar, R. e Kim, J. (2014). Preface to the Special Issue on Intelligent Support for Learning in Groups. *International Journal of Artificial Intelligence in Education*, 1–7.

Levenshtein, V. I. (1965). Binary Codes Capable Of Correcting Deletions, Insertions, And Reversals. Editora Soviet Physics Doklady.

Lin, F. L., Yeh, Y., Hung, Y. H. e Chang, R. I. (2013). Data mining for providing a personalized learning path in creativity: an application of decision trees. *Computer and Education*, 68:199–210.

Mackay, H. (1985). Stimulus equivalence in rudimentary reading and spelling.*Analysis and Intervention in Developmental Disabilities*, 5:373–387.

Markovic, S. e Jovanovic, N. (2012). Learning style as a factor which affects the quality of e-learning. *Artificial Intelligence Review*, 38:303–312.

Millán, E., Descalço, L., Castilho, G., Oliveira, P. e Diogo, S. (2013). Using Bayesian networks to improve knowledge assessment. *Computers and Education*, 60:436–447.

Orlando, A. F. (2009). *Uma Infra-estrutura computacional para o gerenciamento de programas de ensino individualizados*. UFSCar, São Carlos.

Pedri, V. (2012). Web page: http://www.portaleducacao.com.br/educacao/artigos/25184/a-importancia-do-tutor-no-processo-de-aprendizagem-na-modalidade-de-ead . Acessado: 2015-10-05.

Pereira, A. B. C., Marques, L. B., Monteiro, D. C. e Souza, G. N. (2013). A Fuzzy System for Educational Tasks for Children with Reading Disabilities**.** In *The 18th European Conference on Artificial Intelligence, Workshop CIMA - 3rd International Workshop on Combinations of Intelligent Methods and Applications*, p. 57-78, Montpellier, France.

Ronimus, M., Kujala, J. e Tolvanen, Asko. (2014). Children's engagement during digital game-based learning of reading: The effects of time, rewards, and challenge. *Computers & Education*, 71:237–246.

Rothlauf, F. (2006). Representations for genetic and evolutionary algorithms. Editora Verlag Berlin Heidelberg: Springer.

Rose, J., Souza, D. G., Rossito, A. L. e Rose, T. M. S (1989). Aquisição de leitura após história de fracasso escolar: equivalência de estímulos e generalização.*Psicologia: Teoria e Pesquisa*, 5:451-69.

Sampayo, S., Cope, C. J., He, Z. e Byrne, (2013). The effectiveness of adaptive difficulty adjustments on students' motivation and learning in an educational computer game. *Computer& Education*, 69:452-462.

Sharkness, J. (2014). Item Response Theory: Overview, Applications, and Promise for Institutional Research.*New Directions for Institutional Research*, 161: 41–58.

Silva, R. (2015). Web page: http://porvir.org/porpensar/diferenciar-individualizar-personalizar-ensino/20120822. Acessado: 2015-05-09.

Siqueira, E. S. (2012). ALERPG – Jogo digital como recurso complementar para ensino de crianças com Dificuldade de Aprendizagem em Leitura e Escrita. Editora Universidade Federal do Pará, Belém.

Skinner, B. F. (1950). Are theories of learning necessary ?. *The Psychological Review*, 57:196-216.

Souza, D. G., Rose, J. C., Faleiros, T. C., Bortoloti, R., Hanna, E. S. E McIlvane, W. J. (2009). Teaching generative reading via recombination of minimal textual units: a legacy of verbal behavior to children in Brazil. *Revista Internacional de Psicologia y Terapia Psicológica*, 9:19-44.

Souza, G. N., Monteiro, D. C., Pereira. A. B. C., Barros, E. S. e Marques, L. B. (2012). Máquina de aprendizagem como ferramenta de auxílio na análise comportamental no ensino da leitura. *RENOTE - Revista Sobre Novas Tecnologias na Educação*, 10:1-11.

United, Nations. (2015). Web site: http://www.huffingtonpost.com/2014/01/29/kid-literacy-rate_n_4691196.html. Acessado: 2015-25-01.

Verdú, E., Verdú, M. J., Regueras, L. M., Castro, J. P. e Garcia, R (2012). A genetic fuzzy expert system for automatic question classification in a competitive learning environment. *Expert Systems with Applications*, 39:7471–7478.