



UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

TESE DE DOUTORADO

Geração de tarefas de ensino adaptadas através de algoritmos bio-
inspirados para crianças em fase inicial da alfabetização

Gilberto Nerino de Souza Junior

UFPA / ITEC / PPGEE
Campus Universitário do Guamá
Belém-Pará-Brasil
2018

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Gilberto Nerino de Souza Junior

Geração de tarefas de ensino adaptadas através de algoritmos bio-
inspirados para crianças em fase inicial da alfabetização

TD: 12/2018

UFPA / ITEC / PPGEE
Campus Universitário do Guamá
Belém-Pará-Brasil
2018

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Gilberto Nerino de Souza Junior

Geração de tarefas de ensino adaptadas através de algoritmos bio-
inspirados para crianças em fase inicial da alfabetização

Tese submetida à Banca Examinadora do Programa de Pós-graduação em Engenharia Elétrica da UFPA para a obtenção do grau de Doutor em Engenharia Elétrica na área de Computação Aplicada, elaborada sob a orientação do Prof. Dr. Ádamo Lima de Santana e coorientação do Prof. Dr. Dionne Cavalcante Monteiro.

UFPA / ITEC / PPGEE
Campus Universitário do Guamá
Belém-Pará-Brasil
2018

**Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD
Biblioteca do Instituto de Tecnologia / UFPA – Belém-PA**

S729g Souza Junior, Gilberto Nerino de, 1985-
Geração de tarefas de ensino adaptadas
através de algoritmos bio-inspirados para crianças
em fase inicial da alfabetização/ Gilberto Nerino de
Souza Junior. — 2018.
Orientador: Ádamo Lima de Santana;
Coorientador Dionne Cavalcante Monteiro
Tese (Doutorado) – Universidade Federal do Pará,
Instituto de Tecnologia, Programa de Pós-Graduação
em Engenharia Elétrica, Belém, 2018.
1. Inteligência artificial 2. Ensino auxiliado por
computador 3. Algoritmos genéticos 4. Tecnologia
educacional. I. Título.

CDD 23. ed. – 006.3

Elaborado por Maria Ivone Maia da Costa – CRB-2/768

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

GERAÇÃO DE TAREFAS DE ENSINO ADAPTADAS ATRAVÉS DE
ALGORITMOS BIO-INSPIRADOS PARA CRIANÇAS EM FASE INICIAL DA
ALFABETIZAÇÃO

AUTOR: GILBERTO NERINO DE SOUZA JUNIOR

TESE DE DOUTORADO SUBMETIDA À AVALIAÇÃO DA BANCA
EXAMINADORA APROVADA PELO COLEGIADO DO PROGRAMA DE PÓS-
GRADUAÇÃO EM ENGENHARIA ELÉTRICA DA UNIVERSIDADE FEDERAL DO
PARÁ E JULGADA ADEQUADA PARA OBTENÇÃO DO GRAU DE DOUTOR EM
ENGENHARIA ELÉTRICA NA ÁREA DE COMPUTAÇÃO APLICADA NA LINHA
DE PESQUISA EM INTELIGÊNCIA COMPUTACIONAL.

APROVADA EM ____/____/_____

BANCA EXAMINADORA:

Prof. Dr. Ádamo Lima de Santana
(ORIENTADOR - UFPA)

Prof. Dr. Dionne Cavalcante Monteiro
(COORIENTADOR - UFPA)

Prof. Dr. Eloi Luiz Favero
(MEMBRO - UFPA)

Prof. Dr. Jefferson Magalhães de Moraes
(MEMBRO - UFPA)

Prof. Dr. Nandamudi Lankalapalli Vijaykumar
(MEMBRO - INPE)

Prof. Dr. Roberto Celio Limão de Oliveira
(MEMBRO - UFPA)

“Há circunstâncias em que nos entendemos mais facilmente sem palavras”

Edith Stein

AGRADECIMENTOS

Aos amigos dos diversos grupos de pesquisa dos laboratórios LINC (Laboratório de Inteligência Computacional e Pesquisa Operacional) e LAAI (Laboratório de Inteligência Artificial Aplicada), que colaboraram no desenvolvimento de protótipos, algoritmos e auxílios na escrita de artigos. Obrigado amigos, pelos dias de muito trabalho, mas altamente gratificantes, pela dedicação e profissionalismo.

Aos amigos da área da psicologia e pedagogia Leonardo Marques, Paulo Goulart, Myenne Tsutsumi, Sandra Nazaré e Lidyane Braga que foram fundamentais no desenvolvimento desta tese.

Ao meu orientador, Prof. Dr. Ádamo Lima de Santana, cujo ensinamentos e paciência foram determinantes no decorrer do desenvolvimento da pesquisa.

Ao meu coorientador, Prof. Dr. Dionne Cavalcante Monteiro, que buscou sempre me dar ótimos conselhos e ajudar a aprimorar a pesquisa em todos os níveis.

A todos meus familiares e irmãos em Cristo da comunidade católica Kerygma.

A minha irmã Giselle Nerino e meu irmão Rildo Nerino, pela amizade e companheirismo.

A meu pai, Gilberto Nerino, com seu bom humor e sua dedicação. À minha mãe, Maria da Conceição, pelo incentivo e carinho. Obrigado pelos ensinamentos e exemplos meus pais.

Um agradecimento especial à minha esposa, Andrea Araújo Farias, que na sua docilidade, torceu muito para que eu chegasse ao final dessa tese.

Finalmente, agradeço a Deus pela presença que gera de vida, graças e bênçãos na minha família.

SUMÁRIO

1. Introdução	1
1.1. <i>Problemática</i>	5
1.2. <i>Objetivos</i>	6
1.3. <i>Metodologia de pesquisa e desenvolvimento</i>	7
1.4. <i>Equipe</i>	8
1.5. <i>Contribuições</i>	8
1.6. <i>Estrutura da Tese</i>	10
2. Referencial Teórico	11
2.1. <i>Considerações iniciais do capítulo</i>	11
2.2. <i>Análise comportamental e procedimento Matching-to-Sample aplicados no processo de ensino da leitura</i>	11
2.3. <i>Noções sobre a dificuldade de uma tarefa de ensino</i>	18
2.4. <i>Computação bio-inspirada</i>	23
2.4.1. <i>Algoritmos genéticos</i>	24
2.4.2. <i>Algoritmo de otimização por enxame de partículas</i>	27
2.4.3. <i>Algoritmo de otimização por colônia de formigas</i>	29
2.5. <i>Considerações finais do capítulo</i>	32
3. Trabalhos Correlatos	33
3.1. <i>Considerações iniciais do capítulo</i>	33
3.2. <i>Sistemas de ensino adaptáveis</i>	34
3.3. <i>Sistemas de tutoria e aprendizado adaptativos</i>	37
3.4. <i>Sistemas de testes adaptativos</i>	40
3.5. <i>Análise dos trabalhos pesquisados</i>	43
3.6. <i>Considerações finais do capítulo</i>	49
4. Geração de tarefas de ensino adaptadas através de algoritmos bio-inspirados para crianças em fase inicial da alfabetização	50
4.1. <i>Considerações iniciais do capítulo</i>	50
4.2. <i>Sistema para a geração de tarefas</i>	50
4.2.1. <i>Dificuldade das tarefas de ensino</i>	53
4.2.2. <i>Função de fitness</i>	55
4.2.3. <i>Algoritmos bio-inspirados</i>	58
4.2.3.1. <i>Algoritmo Genético para a geração de tarefas</i>	58
4.2.3.2. <i>Otimização por enxame de partículas para a geração de tarefas</i>	60
4.2.3.3. <i>Otimização por colônia de formigas para a geração de tarefas</i>	61

4.2.4. Softwares de aprendizado	61
4.3. Considerações finais do capítulo	64
5. Experimentos e Resultados	65
5.1. Considerações iniciais do capítulo	65
5.2. Síntese dos experimentos e validações realizadas	65
5.3. Validação computacional dos algoritmos bio-inspirados.....	66
5.4. Validação da dificuldade das tarefas por especialistas	71
5.5. Validação das tarefas geradas para o processo de ensino individualizado.....	71
5.6. Engajamento dos softwares de aprendizagem.....	75
5.7. Considerações finais do capítulo	77
6. Conclusão	78
6.1. Contribuições científicas.....	80
6.2. Trabalhos futuros	81
Referências Bibliográficas	83
Apêndices	95
Apêndice I. Algoritmo de ordenação de tarefas baseado no Fluxo de Dificuldade.....	96
Apêndice II. Algoritmo de similaridade entre escolhas textuais.....	97
Apêndice III. Termo de Consentimento Livre e Esclarecido.....	99
Apêndice IV. Imagens suplementares do aplicativo web, “MTS- Player”.....	101
Apêndice V. Imagens suplementares do jogo digital, “As Aventuras de Amaru”.....	102
Apêndice VI. Questionário para validação das dificuldades das tarefas de ensino.....	104
Anexos	108
Anexo A. Parecer Consubstanciado do CEP (Comitê de Ética e Pesquisa).....	109

LISTA DE FIGURAS

FIGURA 2.1 – EXEMPLO DE UM CANAL DE FLUXO.....	20
FIGURA 2.2 – COMPORTAMENTO DA CURVA CARACTERÍSTICA DO ITEM PARA O MODELO DE 1 PARÂMETRO COM DOIS ITENS.	22
FIGURA 2.3 – FLUXOGRAMA DE UM ALGORITMO GENÉTICO.....	24
FIGURA 2.4 – ESQUEMA DE CRUZAMENTO DE UM PONTO.....	26
FIGURA 2.5 - ESQUEMA DE CRUZAMENTO DE DOIS PONTOS.....	26
FIGURA 2.6 - ESQUEMA DE CRUZAMENTO UNIFORME.....	26
FIGURA 2.7 - EXEMPLO DE UM ENXAME DE PARTÍCULAS.	27
FIGURA 2.8 – FLUXOGRAMA DE MOVIMENTAÇÃO DO PSO.....	28
FIGURA 2.9 – REPRESENTAÇÃO DA ESTRUTURA DA POSIÇÃO DE UMA SOLUÇÃO NO ICPSO.	29
FIGURA 2.10 – FLUXOGRAMA DE EXECUÇÃO DE UM ALGORITMO DE ACO.	30
FIGURA 2.11 – EXEMPLO DA ESTRUTURA DE UM GRAFO PARA O ACO.	31
FIGURA 3.12 – ESTRUTURA DE UMA SESSÃO DE ENSINO NO GEIC.....	35
FIGURA 3.13 – UMA CENA DO JOGO DESENVOLVIDO PARA AUXILIAR NO APRENDIZADO DA LEITURA.	35
FIGURA 3.14 – JOGO PARA TREINO DE FONEMAS. O JOGADOR CLICA NA SÍLABA CORRESPONDENTE E UMA ANIMAÇÃO FAZ O CANHÃO ATIRAR NOS BALÕES CLICADOS.	36
FIGURA 4.15 – FLUXOGRAMA GERAL PARA A GERAÇÃO DE TAREFAS ADAPTADAS.....	51
FIGURA 4.16 – EXEMPLO DO MÉTODO PARA CALCULAR A SIMILARIDADE ENTRE ESCOLHAS DE UMA TAREFA. (A) EXEMPLO DE UMA TAREFA BC, ONDE AS SETAS REPRESENTAM A “AÇÃO DE COMPARAÇÃO” E O CÁLCULO DA SIMILARIDADE; (B) O CÁLCULO DO FATOR DE SIMILARIDADE PARA A TAREFA EM QUESTÃO.	54
FIGURA 4.17 – EXEMPLOS DE FLUXO DE DIFICULDADE QUE PODEM SER DEFINIDOS NO SISTEMA.	56
FIGURA 4.18 – EXEMPLO DE UM CONJUNTO DE TAREFAS.	56
FIGURA 4.19 - FLUXO DE DIFICULDADE E A DIFICULDADE DAS TAREFAS.	57
FIGURA 4.20 – (A) ESTRUTURA DE CODIFICAÇÃO DO INDIVÍDUO E (B) EXEMPLO DE UM INDIVÍDUO NO GA.59	
FIGURA 4.21 - EXEMPLO DO MÉTODO DE MUTAÇÃO COM SELEÇÃO SEGUNDO A PROBABILIDADE DE MUTAÇÃO.	59
FIGURA 4.22 - UM EXEMPLO DE UM INDIVÍDUO PARA A REPRESENTAÇÃO DO ICPSO.	60
FIGURA 4.23 - UM EXEMPLO DO GRAFO E DO INDIVÍDUO (LINHA TRACEJADA) NO ACO.....	61
FIGURA 4.24 - APLICATIVO WEB (MTS-PLAYER) PARA O TIPO DE TAREFA CB, COM “TOMATE”(A) , E “BOLO”(B) COMO MODELOS DAS TAREFAS.	62
FIGURA 4.25 - IMAGENS DAS “AVENTURAS DE AMARU”: (A) EXECUÇÃO DE UM MINI-GAME PARA O TIPO DE TAREFA BC, (B) MINI-MAPA.	63
FIGURA 4.26 -ARQUIVO DE CONFIGURAÇÃO CSV DE UM CONJUNTO DE TAREFAS.	64
FIGURA 5.27 -COMPORTAMENTO DA CURVA DE CONVERGÊNCIA DE FITNESS. A LINHA CENTRAL É A MÉDIA E O SOMBREADO É O DESVIO PADRÃO DOS MELHORES INDIVÍDUOS. (A) DESEMPENHO INICIAL, (B) DESEMPENHO INTERMEDIÁRIO.....	68
FIGURA 5.28 -DIFICULDADE DE TAREFAS GERADAS PELO PERFIL DE DESEMPENHO (A) INICIAL E (B) INTERMEDIÁRIO.	69

FIGURA 5.29 – DESENHO DA ATIVIDADE INDIVIDUALIZADA.....	72
FIGURA 5.30 – ATIVIDADE DE ENSINO INDIVIDUAL PARA DOIS ALUNOS.	74
FIGURA 5.31 – APLICAÇÃO DA SESSÃO DE ENSINO EM UM LABORATÓRIO DE COMPUTAÇÃO.	75
FIGURA 5.32 – DESENHO DA ATIVIDADE PARA A TURMA PARA A COLETA DO ENGAJAMENTO DOS ALUNOS.	76
FIGURA AP-33 – ÁREA DE INSERÇÃO DE TAREFAS DO MTS-PLAYER.....	101
FIGURA AP-34 – ÁREA DE EDIÇÃO DE TAREFAS DO MTS-PLAYER.....	101
FIGURA AP-35 – TAREFA SENDO APRESENTADA NO MTS-PLAYER.....	101
FIGURA AP-36 – INÍCIO DA HISTÓRIA DO JOGO EM FORMATO DE GRAVURAS.....	102
FIGURA AP-37 – TUTORIAL NO INÍCIO DO JOGO.	102
FIGURA AP-38 – ÍTENS COLETÁVEIS ENTRE AS TAREFAS DO JOGO.	103
FIGURA AP-39 – EXECUÇÃO DAS TAREFAS EM MINI-GAMES.....	103

LISTA DE TABELAS

TABELA 1.1 – EQUIPE DE PESQUISA.	8
TABELA 2.2 – EXEMPLOS DE TIPOS DE RELAÇÕES DE TAREFAS DO PROCEDIMENTO MTS.....	16
TABELA 2.3 – TRÊS EXEMPLOS DE DIFERENTES RELAÇÕES ENTRE TAREFAS DO TIPO CRMTS.....	17
TABELA 3.4 – SUMÁRIO DE TRABALHOS ANALISADOS. SISTEMAS DE ENSINO ADAPTÁVEIS: SISTEMAS EDUCACIONAIS AUXILIADOS POR UM ESPECIALISTA. SISTEMAS DE TUTORIA E APRENDIZADO ADAPTATIVOS: SISTEMAS EDUCACIONAIS AUXILIADOS POR TÉCNICAS DE INTELIGÊNCIA COMPUTACIONAL. SISTEMAS DE TESTES ADAPTATIVOS: SISTEMAS EDUCACIONAIS FOCADOS NA AVALIAÇÃO DO APRENDIZADO AUXILIADOS COM TÉCNICAS DE INTELIGÊNCIA COMPUTACIONAL.....	44
TABELA 4.5 – EXEMPLO DE UM CONJUNTO DE TAREFAS DE ENSINO. ID = IDENTIFICADOR NUMÉRICO DA TAREFA.....	52
TABELA 4.6 – DESCRIÇÃO DAS RESTRIÇÕES E SUAS PENALIDADES ATRIBUÍDAS AO VALOR DE <i>FITNESS</i>	58
TABELA 5.7 – CONFIGURAÇÃO PARA A GERAÇÃO DE TAREFAS.	67
TABELA 5.8 – CONFIGURAÇÃO DO GA, ICPSO E ACO.....	67
TABELA 5.9 – UMA AMOSTRA DE TAREFAS GERADAS EM ORDEM DE APRESENTAÇÃO PARA OS ALUNOS PARA O PERFIL DE DESEMPENHO INICIAL.	70
TABELA 5.10 – UMA AMOSTRA DE TAREFAS GERADAS EM ORDEM DE APRESENTAÇÃO PARA OS ALUNOS PARA O PERFIL DE DESEMPENHO INTERMEDIÁRIO.	70
TABELA 5.11 – RESULTADO DA ENTREVISTA SEMI-ESTRUTURADA.....	74
TABELA 5.12 – NÚMERO DE REPETIÇÕES DE ESTÁGIOS.	74
TABELA 5.13 – NÚMERO GERAL DE RESPOSTAS CORRETAS DO PRÉ-TESTE E PÓS-TESTE (NÚMERO TOTAL DE 36 TAREFAS PARA CADA ALUNO).	75
TABELA 5.14 – PORCENTAGEM DE RESPOSTAS POSITIVAS NOS DOIS DIAS DO EXPERIMENTO.	76
TABELA APII.15 – MÉDIA DA INDICAÇÃO DA OPINIÃO DOS ESPECIALISTAS ENTRE PALAVRAS TEXTUAIS PARA 15 PALAVRAS. QUANTO MAIOR O VALOR, MAIOR A SIMILARIDADE.	98

LISTA DE SIGLAS

ACM	<i>Association for Computing Machinery</i>
ACO	<i>Ant Colony Optimization</i>
ALEPP	<i>Aprendendo a Ler e Escrever em Pequenos Passos</i>
CAT	<i>Computerized Adaptive Testing</i>
CCI	<i>Curva Característica do Item</i>
CEP	<i>Comitê de Ética e Pesquisa</i>
CR	<i>Constructed Response</i>
CRMTS	<i>Constructed Response - Matching-to-Sample</i>
CSV	<i>Comma-Separated Values</i>
ERIC	<i>Education Resources Information Center</i>
GA	<i>Genetic Algorithm</i>
GEIC	<i>Gerenciador de Ensino Individualizado por Computador</i>
IA	<i>Inteligência Artificial</i>
IAED	<i>Inteligência Artificial na educação</i>
IC	<i>Inteligência Computacional</i>
ICPSO	<i>Integer and Categorical Particle Swarm Optimization</i>
IBGE	<i>Instituto Brasileiro de Geografia e Estatística</i>
IEEE	<i>Institute of Electrical and Electronic Engineers</i>
ITS	<i>Intelligent Tutoring Systems</i>
LAAI	<i>Laboratory of applied artificial intelligence</i>
LINC	<i>Laboratório de inteligência computacional e pesquisa operacional</i>
MTS	<i>Matching-to-Sample</i>
NP-	<i>Non-deterministic Polynomial time</i>
PSO	<i>Particle Swarm Optimization</i>
RMSE	<i>Root Mean Square Error</i>
TAC	<i>Testes Adaptativos Computadorizados</i>
TRI	<i>Teoria de Resposta ao Item</i>
TTC	<i>Teoria de testes clássico</i>
UFSCar	<i>Universidade Federal de São Carlos</i>
UFPA	<i>Universidade Federal do Pará</i>
UNESCO	<i>United Nations Educational, Scientific and Cultural Organization</i>

LISTA DE SÍMBOLOS

i, j, k	São os índices de elementos em vetores, listas ou matrizes
T_C	Número de combinações de diferentes conjuntos de tarefas
N_{max}	Número máximo de escolhas utilizadas em uma sessão de ensino
$N_{C_{max}}$	Número máximo de escolhas em uma tarefa
$N_{C_{min}}$	Número mínimo de escolhas em uma tarefa
T_{max}	Número máximo de tipos de tarefas em uma sessão de ensino
M_{max}	Número máximo de modelos (palavras a serem ensinadas)
N_T	Número de tarefas em um conjunto de tarefas usado em uma sessão de ensino
U_{ij}	Variável que assume valores 1 quando uma pessoa j responde corretamente o item i . Variável assume valor 0 caso uma pessoa j não responda corretamente o item i
θ_j	É o conhecimento ou habilidade da pessoa j
b_i	É o parâmetro relativo à dificuldade do item i
$P(U_{ij} \theta_j; b_i)$	Probabilidade da pessoa j com conhecimento θ responder corretamente o item i , com dificuldade b_i
a_i	Parâmetro de inclinação discriminativo do item i
c_i	Parâmetro de probabilidade de acerto ao acaso do item i
e	É a constante de valor é 2,718, base dos logaritmos neperianos
V_i	Velocidade da partícula i
X_i	Posição da partícula i
w	É o parâmetro de inercia
c_1	É o parâmetro cognitivo
c_2	É o parâmetro social
r_1, r_2	São variáveis aleatórias com valores 0 ou 1
M_i	É a melhor posição local encontrada pela partícula i
Mg	É a melhor posição global encontrada pelo enxame de partículas
X_i	Representação da posição da partícula i para o ICPSO
$Y_{i,j}$	É a distribuição de probabilidades j da partícula i
$y_{i,j,k}$	É a probabilidade do elemento k , de uma distribuição de probabilidade j e de uma partícula i
n, m	Número de elementos de um vetor ou lista
l	É um nó em um grafo
x, z	São os índices do nó de partida de uma aresta
x', z'	São índices do nó de chegada da uma aresta
$[l_{(x,z)}, l_{(x',z')}]$	A aresta partindo do nó $l_{(x,z)}$ e chegando ao nó $l_{(x',z')}$
$q_{[l_{(x,z)}, l_{(x',z')}]^k}$	É a probabilidade da formiga k escolher a aresta entre os nós $l_{(x,z)}$ e $l_{(x',z')}$.
τ	Quantidade de feromônio depositado em uma aresta
$\tau_{[l_{(x,z)}, l_{(x',z')}]}$	É que quantidade de feromônio depositado na aresta entre os nós $l_{(x,z)}$ e $l_{(x',z')}$.
$\Delta\tau_{[l_{(x,z)}, l_{(x',z')}]^k}$	É que quantidade de feromônio depositado na aresta entre os nós $l_{(x,z)}$ e $l_{(x',z')}$ pela formiga k .
ρ	É a taxa de evaporação do feromônio
f	Valor de fitness (aptidão) de uma solução
D	É a dificuldade adaptativa de uma tarefa

κ	<i>Índice dos tipos de tarefas</i>
η	<i>Índice dos intervalos de similaridade da tarefa</i>
I1, I2, I3, I4 e I5	<i>São os intervalos de similaridade possíveis com valores entre [0, 0,2[, [0,2, 0,4 [, [0,4, 0,6 [, [0,6, 0,8 [, [0,8, 1], respectivamente.</i>
βt_{κ}	<i>Fator de dificuldade do tipo de tarefa κ</i>
βs	<i>Fator de similaridade da tarefa</i>
βs_{η}	<i>Fator de dificuldade de similaridade tarefa no intervalo η</i>
θt_{κ}	<i>Conhecimento do aluno para o tipo de tarefa κ</i>
θs_{η}	<i>Conhecimento do aluno para o intervalo de similaridade η</i>
n_c	<i>É o número de opções da tarefa</i>
s_i	<i>É a similaridade entre o modelo e a escolha i</i>
$Q t_{\kappa}$	<i>É o número de tarefas respondidas do tipo de tarefa κ</i>
$C t_{\kappa}$	<i>É o número de tarefas corretamente respondidas do tipo de tarefa κ</i>
$Q s_{\eta}$	<i>É o número de tarefas respondidas para os intervalos de similaridade η</i>
$C s_{\eta}$	<i>É o número de tarefas respondidas corretamente para os intervalos de similaridade η</i>
r_{min}, r_{max}	<i>É o intervalo máximo e mínimo usados para normalizar θt_{κ} e θs_{η}</i>
T_i	<i>Representa uma tarefa i</i>
\mathbf{D}	<i>Vetor de dificuldades de um conjunto de tarefas</i>
\mathbf{F}	<i>Fluxo de dificuldade que é um vetor de valores de dificuldade inseridos pelo tutor ou gerado por uma função</i>
F_i	<i>Um valor de dificuldade para uma tarefa i, inserido pelo tutor ou gerado por uma função</i>
$A(\mathbf{D}, \mathbf{F})$	<i>Área entre os vetores \mathbf{D} e \mathbf{F}.</i>
Φ_j	<i>Valor de penalidade do índice j</i>
n_R	<i>É o número de tarefas idênticas em um conjunto de tarefas</i>
n_{TM}	<i>É o número de tarefas com tipo de tarefas e modelos idênticos em um conjunto de tarefas</i>
n_M	<i>É o número de tarefas com modelos idênticos</i>
n_T	<i>É o número de tarefas com tipos de tarefas idênticos</i>
n_a	<i>É o número de atributos da tarefa, onde os atributos são o tipo de tarefa, o modelo e a quantidade de escolhas.</i>
p'	<i>É a nova probabilidade para a próxima iteração do operador genético</i>
p	<i>É a probabilidade atual do operador genético</i>
d	<i>É a taxa de diversidade da população atual</i>
d_t	<i>É o alvo da diversidade da população</i>
ξ	<i>Constante de sensibilidade</i>
p_{min}, p_{max}	<i>São os limites mínimo e máximo de probabilidade dentro do intervalo de [0,1]</i>

RESUMO

Avanços em sistemas de aprendizagem ao longo das últimas duas décadas permitiram o desenvolvimento de tecnologias que auxiliam no engajamento de alunos. Embora esses sistemas possam usar procedimentos comportamentais para melhorar as habilidades em leitura, melhores resultados para cada aluno são obtidos na elaboração manual de um conjunto de tarefas por um especialista educacional. Todavia, o uso de um processo manual acaba acarretando demasiado tempo, esforço e subjetividade para a criação das tarefas de ensino. Adicionalmente, a geração automática de tarefas para o ensino da leitura pode ser inviável devido ao alto espaço de busca das possíveis combinações de tarefas. Este processo poderia considerar a adaptação da dificuldade de uma tarefa ao conhecimento do aluno, algo pouco explorado em trabalhos educacionais para crianças no início do aprendizado da leitura. A presente Tese apresenta uma abordagem para gerar tarefas do procedimento *Matching-to-Sample* para o ensino da leitura, adaptando suas dificuldades através de meta-heurísticas de otimização bio-inspiradas. Esta abordagem utiliza-se de resultados de pré-testes aplicados a alunos e da configuração de conteúdos de ensino determinados por tutores educacionais; esses dados permitem a utilização dos algoritmos de geração de tarefas e em seguida as tarefas podem ser apresentadas em softwares de aprendizagem. Experimentos demonstraram uma melhor convergência do algoritmo genético para este domínio, sendo que este algoritmo foi capaz de gerar tarefas em um nível de dificuldade adaptadas aos alunos e de acordo com pré-testes e configurações de atributos das tarefas definidas por psicólogos comportamentais. Como validação para este estudo, as tarefas foram aplicadas a um grupo de alunos em estágios iniciais da alfabetização obtendo efeitos satisfatórios no processo individual de aprendizagem. Adicionalmente foram implementados dois softwares interativos de de aprendizagem, um por meio do jogo digital e o outro por um aplicativo web, onde o uso do jogo digital com características lúdicas demonstrou aceitação superior no uso de tarefas de ensino adaptadas para crianças em fase inicial da alfabetização.

PALAVRAS-CHAVE: Geração de tarefas, Procedimento *Matching-to-Sample*, Dificuldade adaptada, Algoritmos bio-inspirados, Softwares de aprendizado, Ensino da leitura

ABSTRACT

Advances in learning systems over the past two decades have enabled the development of technologies that help in the engagement of students. Although these systems may use behavioral procedures to improve reading skills, better outcomes for each student are obtained in the manual elaboration of a set of tasks by educational experts. However, the use of a manual process requires too much time, effort and subjectivity for the creation of tasks. Additionally, even with the aid of computational processes, the automatic generation may be impracticable due to the high search space for the possible combinations of tasks. This process could consider adapting the difficulty of a task to the student's knowledge, something little explored in educational work for children at the beginning of reading learning. The present thesis implements an approach to generate teaching tasks from the Matching-to-Sample procedure, adapting its difficulties through bio-inspired optimization meta-heuristics. This approach uses pre-test results applied to students and the configuration of teaching contents determined by educational tutors; these data allow the use of the algorithms to generate tasks and then the tasks can be presented in learning software. Experiments demonstrated a better convergence of the genetic algorithms for this domain, being able to generate tasks on a level of difficulty adapted to the students, and also according to pretests and configurations of attributes of the tasks defined by behavioral psychologists. As validation for this study, the tasks were applied to a group of students in the early stages of literacy achieving satisfactory effects in the individual learning process. In addition, two interactive learning software were implemented through a digital game and a web application, where the use of the digital game with playful features showed superior acceptance in the use of teaching tasks adapted for children in the initial phase of literacy.

KEYWORDS: Task Generation, Matching-to-Sample Procedure, Adapted Difficulty, Bio-Inspired Algorithms, Learning Software, Teaching of the Reading.

1. Introdução

Avanços em sistemas de aprendizado e tutoria inteligentes ao longo das últimas duas décadas permitiram o desenvolvimento e a implantação de tecnologias de aprendizagem altamente eficazes em diversos domínios, que podem auxiliar professores e alunos em diversos grupos etários (Kumar & Kim, 2014). Vários serviços educacionais difundem sistemas que permitem o acesso a centenas de horas de conteúdo e em diversos formatos de dados, onde inúmeros elementos interativos são estudados para enriquecer fatores motivacionais e ajudar no ensino-aprendizado (Prensky, 2003). Exemplos destes ambientes vão desde cursos online, sistemas baseados na web, plataformas de redes sociais cooperativas/competitivas e jogos educativos que ajudam na satisfação do usuário (Cabada, Barrón Estrada, & Reyes García, 2011; L. X. Chen & Sun, 2016; Hwang & Wang, 2016; G. G. Smith et al., 2013; Verdú, Verdú, Regueras, De Castro, & García, 2012).

Apesar deste avanço tecnológico, segundo o último relatório da UNESCO (*United Nations Educational, Scientific and Cultural Organization*), publicado em 2014, aproximadamente 250 milhões de 650 milhões de crianças em idade escolar primária não sabem ler (UNESCO, 2012, 2014). Nos Estados Unidos da América, cerca de 16% dos alunos em fases iniciais do aprendizado tem problemas com a leitura (Mueller, Olmi, & Saunders, 2000). Já no Brasil, mais de 4 milhões de crianças entre 5 e 7 anos não estão alfabetizadas, segundo os dados do IBGE (Instituto Brasileiro de Geografia e Estatística) de 2011 (IBGE, 2011). O dado da Avaliação Nacional de Alfabetização mostra que 50% das crianças entre 5 e 8 anos não tem conhecimento suficiente de leitura, e na região norte do Brasil a situação é pior, pois 70,21% dos estudantes tem desempenho insuficiente na leitura de pequenos textos (Bermúdez, 2017).

Em quesitos monetários, estima-se um prejuízo anual por volta de 129 bilhões de dólares para os governos ao redor do mundo (The Huffington Post, 2014), pois o gasto com educação deve ser realocado e incrementado para não prejudicar o futuro de uma geração. Este cenário sugere que novos métodos podem ser utilizados para melhorar e acelerar o processo de aprendizado da leitura, em especial para crianças que possuem limitação de aprendizado.

Alguns pesquisadores afirmam que o processo de aprendizado da leitura consiste em aspectos simbólicos e comportamentais que devem ser estabelecidos pelas crianças

(Hanna, Karino, Araújo, & Souza, 2010; Mueller et al., 2000), e a utilização dos conceitos inerentes da análise comportamental e equivalência de estímulos podem auxiliar no aumento do vocabulário (Galvão & Barros, 2001; Rodrigo, 2009; J. C. de Rose & Bortoloti, 2007; Todorov, 1982). Neste contexto, o procedimento *Matching-to-Sample* (MTS), ou emparelhamento ao modelo, é um procedimento da análise comportamental que define comportamentos simbólicos como meio de estabelecer reciprocidade entre elementos e suas propriedades (J. C. de Rose & Bortoloti, 2007).

Para descrever uma situação onde busca-se estabelecer reciprocidade entre elementos, destaca-se o seguinte exemplo. Quando uma criança está aprendendo a falar é comum que os pais mostrem à criança um elemento e depois digam à criança o nome deste elemento. Ao ensinar a criança a falar “cachorro”, e possivelmente este animal se aproxima, os pais podem apontar e dizer “cachorro”. Quando a criança começa a falar a palavra cachorro, os pais podem parabenizar a criança. Depois disso, é muito provável que a criança diga novamente a palavra “cachorro” quando o animal se aproximar. O comportamento dessa criança pode ser estabelecido diante de outros estímulos que possuem características comuns (imagens e efeitos sonoros que representem um cachorro). Quando isso ocorre, os estímulos são considerados equivalentes entre si, ou seja, uma imagem de cachorro e o som da palavra falada “cachorro”, refere-se a um mesmo elemento.

O MTS pode ser usado para ensinar através do paradigma de equivalência de estímulos a fim conferir reciprocidade entre elementos. A equivalência de elementos A, B e C pode ser constatada se um elemento A tem uma relação com um elemento B, e se um elemento B tem relação com um elemento C (Goyos, 2012). Para que um aluno seja capaz de responder “A é igual a C”, tarefas MTS são apresentadas repetidas vezes a alunos apresentando estímulos e *feedbacks* (reforçadores de aprendizado caso o aluno responda corretamente uma tarefa), estabelecendo a reciprocidade entre estímulos.

Este procedimento é amplamente utilizado para aquisição de aprendizado básico em diversos ramos do conhecimento (Dixon, Belisle, Stanley, Daar, & Williams, 2016; Walker & Rehfeldt, 2012). No domínio da análise comportamental, as tarefas do procedimento MTS podem ser pré-organizadas e projetadas para ajudar as crianças a associar um estímulo de referência a outro tipo diferente de estímulo (Hamari & Koivisto, 2014; J.-C. Hong et al., 2013).

Alguns programas de ensino utilizam o procedimento MTS para auxiliar alunos na aquisição da escrita e leitura, com ou sem limitação de aprendizado (J. Rose, Souza, Rossito, & Rose, 2012). Porém, a geração de tarefas de ensino é feita de modo manual, ou seja, um tutor deve criar as tarefas baseado na sua experiência, observando as características individuais de aprendizado do aluno em responder corretamente uma tarefa. Este processo acaba por inerentemente acarretar demasiado tempo, esforço e subjetividade para a criação de um repertório de tarefas de ensino.

Além disso, o grau de dificuldade de uma tarefa é um fator de importância que deve ser levado em conta. A dificuldade para se realizar uma tarefa pode tornar o processo de ensino e engajamento mais cansativo, irritante ou frustrante, já que tarefas muito fáceis ou muito difíceis podem interferir no processo de aprendizagem e na motivação do aluno (De-Marcos, Domínguez, Saenz-De-Navarrete, & Pagés, 2014; Ronimus & Richardson, 2014; Sampayo-Vargas, Cope, He, & Byrne, 2013). Segundo Conejo *et al.* (2014), uma tarefa individualizada deve passar por ajustes de dificuldade para garantir que o engajamento do aluno seja mantido. Neste caso, o fluxo de ensino é influenciado pela dificuldade da sequência em que as tarefas são apresentadas aos alunos.

Em consequência, o grau de dificuldade de cada tarefa pode ser personalizado, permitindo ainda ao instrutor, o controle suficiente para ajustar o fluxo de ensino. De acordo com Chang & Ke (2013), a personalização do conteúdo de ensino é um problema de otimização combinatória que pode ser resolvido com meta-heurísticas apropriadas. Essas meta-heurísticas podem ser ajustadas a diferentes problemas de otimização, para que possam explorar um grande espaço de busca e ter a capacidade de chegar a uma solução satisfatória (Christudas, Kirubakaran, & Thangaiah, 2016; Izakian, Ladani, Abraham, & Snášel, 2010; Krynicki, Jaen, & Navarro, 2016).

No contexto da geração de tarefas, alguns sistemas utilizam técnicas computacionais para o ensino em diversos ramos do conhecimento, mas estes métodos não são projetados para gerar um conjunto de tarefas MTS. Adicionalmente, no contexto do procedimento comportamental MTS, a geração de tarefas com dificuldade adaptadas foi pouco explorada no processo de ensino-aprendizagem de crianças no início da alfabetização como cita o trabalho de Pereira *et al.* (2012).

Na presente tese foram utilizadas meta-heurísticas bio-inspiradas para gerar tarefas do procedimento MTS para o ensino da leitura. Até onde sabe-se, não existem estratégias algorítmicas que usem as vantagens de algoritmos bio-inspirados para gerar

tarefas MTS. Diante disso, para preencher adequadamente essa lacuna, foi estimado a categoria mais adequada de meta-heurísticas para este trabalho. Utilizou-se três classes principais de algoritmos bio-inspiradas: algoritmos genéticos (GA - *Genetic Algorithm*), algoritmo de otimização por enxame de partículas (PSO - *Particle Swarm Optimization*) e algoritmo de otimização por colônia de formigas (ACO - *Ant Colony Optimization*). Estas meta-heurísticas se mostraram apropriadas para serem utilizadas para o problema de geração de tarefas devido o problema ser uma busca combinatorial e apresentar diversas possibilidades de soluções. Estes algoritmos bio-inspirados são focados para a resolução destes tipos de problemas que geralmente possuem máximos locais, mínimos locais, máximos globais e mínimos globais.

Como o problema de geração de tarefas baseado em procedimentos MTS requer que as variáveis sejam representadas em um formato discreto, os algoritmos utilizados neste trabalho buscam transpor esta limitação, pois estes algoritmos possibilitam o uso de variáveis discretas e categóricas. Também neste trabalho, as tarefas geradas por estes algoritmos são individualizadas através de um modelo de dificuldade adaptativo, que estima a probabilidade de o aluno responder incorretamente a uma determinada tarefa.

Verificar o engajamento de alunos em tarefas MTS também é um fator importante (Marques et al., 2013). Por esta razão, buscou-se analisar o engajamento do aluno diante de um conjunto de tarefas geradas automaticamente. Para isto, dois softwares de aprendizado foram especialmente desenvolvidos neste estudo: um aplicativo web, com ênfase na apresentação de tarefas convencionais; e um jogo digital, para aplicar características lúdicas na apresentação de tarefas.

Para analisar a aplicabilidade dessa abordagem, diversos experimentos e atividades como estudos de caso foram realizados. Validação entre os algoritmos bio-inspirados, identificação da dificuldade das tarefas de ensino por especialistas, e aplicação das tarefas de ensino para grupos de alunos foram efetivadas.

No total foram 9 especialistas consultados e 40 alunos, crianças entre 5 e 7 anos de idade e em fase inicial de aprendizagem, que participaram das atividades de ensino do vocabulário português-brasileiro. Pré-testes e pós-testes foram aplicados para coletar dados sobre aprendizagem, e entrevistas semi-estruturadas foram conduzidas com os alunos para descobrir o engajamento das tarefas geradas pelo algoritmo genético modelado, que foi o algoritmo que se revelou mais estável nesta pesquisa.

Portanto, com os experimentos realizados neste estudo, foi possível avaliar a abordagem de geração de tarefas para os alunos nas fases iniciais de alfabetização e verificar a aplicabilidade, tanto em ambientes de ensino convencionais quanto lúdicos de ensino.

1.1. Problemática

A subjetividade na criação das tarefas é uma característica existente na criação das tarefas por um agente humano, que varia de um tutor educacional em relação a outro tutor educacional. Um conjunto de tarefas criado por um tutor, que visou ensinar um conteúdo para um aluno individualmente, pode ter tarefas diferentes quando criado por outro tutor, mesmo se o objetivo foi ensinar o mesmo conteúdo para o mesmo aluno. Esta possibilidade mostra que existe uma subjetividade na criação de tarefas. Levando em consideração que um profissional educacional pode ser um professor de ensino básico, um psicólogo pesquisador ou um pedagogo, as possibilidades de tarefas criadas por um agente humano podem naturalmente divergir.

Logo, o problema na geração de um conjunto de tarefas de ensino da leitura no escopo desta tese se baseia na necessidade de determinar os valores de dificuldade de cada tarefa e adaptar esses valores para cada aluno. Ao realizar isso automaticamente, o tutor pode configurar um conjunto de valores de dificuldade e, com o auxílio de algoritmos computacionais, criar tarefas mais adequadas para o aluno.

Neste contexto, quando os tutores definem um fluxo de dificuldade para o ensino, eles estão definindo a sequência de dificuldades de tarefas que consideram apropriadas. Usando processos computacionais ingênuos (por exemplo, força bruta, caminhada aleatória ou busca gananciosa) é possível ajudar o instrutor a gerar um conjunto de tarefas MTS, e depois estimar a dificuldade de cada uma. No entanto, esses métodos não são ideais, pois exige a execução de um número excessivamente grande de combinações possíveis de tarefas. Para resolver esse problema, é necessário combinar diferentes atributos, como palavras, número de escolhas e tipos de tarefas para gerar um conjunto de tarefas. Essa combinação produz uma ampla gama de tarefas baseadas na Equação 1.1.

$$T_C = \left(\sum_{N_{Cmin}}^{N_{Cmax}} \left(\frac{N_{max}!}{N_{Cmin}!(N_{Cmax}-N_{Cmin})!} \right) \right) * T_{max} * M_{max} * N_T \quad (1.1)$$

Onde:

T_C : é o número de combinações possíveis de diferentes conjuntos de tarefas;

N_{max} : é o número máximo de escolhas usadas na sessão de treinamento;

$N_{C_{max}}$: é o número máximo de escolhas em uma tarefa;

$N_{C_{min}}$: é o número mínimo de escolhas em uma tarefa;

T_{max} : é o número máximo de tipos de tarefas;

M_{max} : é o número máximo de modelos (palavras a serem ensinadas);

N_T : é o número de tarefas em um conjunto usado para a sessão de ensino.

Para ilustrar o uso da Equação 1.1 em unidades de ensino, considere os seguintes valores geralmente utilizados em uma sessão de ensino de (J. Rose et al., 2012): $N_{max} = 80$, $N_{C_{max}} = 5$, $N_{C_{min}} = 2$, $T_{max} = 6$, $M_{max} = 40$ e $N_T = 36$. O resultado é aproximadamente $T_C = 2,22 \times 10^{11}$ combinações possíveis de tarefas. Esta abordagem seria inviável para busca por força bruta em computadores comuns ou dispositivos móveis usados nas escolas¹.

Além disso, a possibilidade de usar esses processos computacionais ingênuos não garantiria que uma tarefa difícil para um aluno X seja igualmente difícil para um aluno Y. É provável que uma mesma solução, ou seja, um conjunto de tarefas, tenha valores de aptidão diferentes. Isto ocorre pois existe uma dependência da configuração de ensino (palavras a serem aprendidas, tipos de tarefas e conhecimento do aluno no conteúdo), dificultando que os métodos de busca separem os mínimos locais dos globais. Para tanto, os algoritmos propostos nesta pesquisa buscam aproximar as dificuldades das tarefas adaptadas a um fluxo de dificuldade (fluxo de ensino) inserido pelo tutor educacional, reduzindo a carga subjetiva e o custo para a seleção das tarefas.

1.2. Objetivos

O objetivo geral dessa tese é implementar uma abordagem para a geração de tarefas com dificuldade adaptadas ao ensino da leitura, utilizando o procedimento MTS e com mecanismos de busca executados através de algoritmos bio-inspirados, a fim de

¹ A abordagem por força bruta foi testada em um computador desktop Intel Core I7, 2.70 GHz, sendo interrompida após 72 horas de execução não produzindo resultado satisfatório.

produzirem o melhor conjunto de tarefas de ensino para alunos em fase inicial da alfabetização.

Os objetivos específicos definem as particularidades em torno deste estudo e complementam o desafio do objetivo principal, destacados como:

- Criação de um modelo de dificuldade de tarefas adaptadas com base no procedimento MTS;
- Modelagem e comparação de algoritmos bio-inspirados para a geração de tarefas de ensino adaptadas e próximo ao conteúdo de ensino inserido pelo tutor educacional;
- Implementação de softwares de aprendizado lúdicos e convencionais, para crianças entre 5 e 7 anos de idade, possibilitando um melhor engajamento no aprendizado.

O produto final deste trabalho pretende ajudar psicólogos e educadores que trabalham com crianças no processo de ensino aprendizagem.

1.3. Metodologia de pesquisa e desenvolvimento

Para alcançar os objetivos desta tese, foi feita a pesquisa de trabalhos que se enquadrem ao contexto da análise comportamental e na geração automática de tarefas de ensino envolvendo habilidades individuais e dificuldade de uma tarefa de ensino.

Conseqüentemente, viu-se nesta lacuna uma motivação para a proposição de um método capaz de facilitar o processo de geração de tarefas para o ensino e avaliação do aprendizado por parte do tutor, professor ou psicólogo.

Logo, a pesquisa foi guiada para identificar a interseção entre os trabalhos correlatos relacionados a: sistemas adaptados, sistemas de aprendizado e tutoria inteligentes, e testes adaptativos computadorizados.

No âmbito de técnicas de inteligência computacional, é proposta a utilização de métodos vindouros da computação bio-inspirada, visto que o problema abordado pode ser considerado como de busca e otimização. Algoritmos genéticos, métodos de inteligência por enxame e otimizações por colônia de formiga foram pesquisados e modelados a fim de realizar a busca de tarefas próximas ao fluxo de ensino proposto pelos tutores. Por fim, foi sugerido que um sistema especialista seja experimentado em ambientes reais de aprendizado, tanto convencional quanto motivacional para o ensino.

1.4. Equipe

A presente pesquisa contou com o auxílio de uma equipe multidisciplinar, o qual não poderia ser realizado sem a qualidade e profissionalismo que cada participante exerceu. Algumas pessoas tiveram uma participação curta durante o desenvolvimento dessa tese e outras permaneceram, na equipe do início até o seu término, mas todos tiveram um papel significativo neste trabalho. Na Tabela 1.1 são expostos os nomes dos participantes desta equipe, bem como o papel que desempenhou durante o período deste trabalho.

Tabela 1.1 – Equipe de pesquisa.

Nome	Papel durante a execução do trabalho
Ádamo Lima de Santana	Orientador e pesquisador
Dionne Cavalcante Monteiro	Coorientador e pesquisador
Leonardo Marques	Pesquisador e analista comportamental
Paulo Goulart	Pesquisador e analista comportamental
Myenne Tsutsumi	Pesquisadora e psicóloga
Igor Araújo	Pesquisador e engenheiro da computação
Vincent Tadaiesky	Pesquisador e engenheiro da computação
Marcia Fontes	Pesquisadora e engenheira da computação
Antonio Fernando Jacob	Pesquisador e Cientista da computação
Lidyane Braga Albim	Comunicadora social e radialista
Sandra Nazaré Parente	Diretora e coordenadora pedagógica
Daniel Felipe	Engenheiro da computação e programador
Jefferson Pantoja	Engenheiro da computação e programador
André Fernandes	Engenheiro da computação e programador
Paulo Oliveira	Engenheiro da computação e programador
Pedro Salvador	Engenheiro da computação e programador
Elton Sales	Cientista da computação e pesquisador
Yvan Pereira	Cientista da computação e programador
Felipe Eliasquevici	Cientista da computação e programador
Giordanna de Gregoriis	Cientista da computação e artista gráfico
Hyago Pinheiro	Cientista da computação e programador
Abner Cardoso	Graduando em Ciência da Computação e programador
Francielma Assunção	Graduanda em Sistemas de Informação e pedagoga
Jefferson Luís	Graduando em engenharia da computação e programador

1.5. Contribuições

Os trabalhos prévios sobre a criação de tarefas de ensino via procedimento de *Matching-To-Sample* eram baseados na criação manual de tarefas, onde o profissional de ensino, seja ele um psicólogo ou pedagogo, necessitaria selecionar todos os estímulos, agrupa-los e depois encadeá-los em sequência. Além disso, muitas vezes essa seleção de tarefas é subjetiva de tutor para tutor, não considerando as aptidões de cada aluno. Nesse sentido, a abordagem presente nesta tese contribui com as funcionalidades descritas a seguir:

- Viabiliza a geração de um conjunto completo de tarefas de ensino para o procedimento MTS, visto que ainda não existia uma abordagem que pudesse realizar a geração de tarefas de maneira individualizada e seguindo um fluxo de ensino.
- Viabiliza a aplicação de uma abordagem que engloba: a aplicação de um pré-teste; a configuração de ensino; a geração de tarefas; e o ensino em softwares de aprendizagem com as tarefas geradas. Isto possibilita a sua utilização em ambientes reais de ensino, como salas de aulas informatizadas.
- Propõe um modelo que estima a dificuldade de uma tarefa do procedimento MTS e adaptada ao aluno.
- Disponibiliza a modelagem de um algoritmo bio-inspirado apto para a geração de tarefas adaptadas e obedecendo a configuração de ensino, configurada por um especialista na área da análise comportamental.
- Permite o ensino individualizado, que respeita o modo de aprendizado de cada aluno, viabilizando um melhor ensino durante o processo de alfabetização.
- Dispõe o resultado de um estudo sobre engajamento e satisfação, utilizando tarefas adaptadas computacionalmente e apresentadas em softwares lúdicos e convencionais de aprendizado.

Este doutorado também produziu contribuições científicas por meio de publicações de artigos científicos em conferências, congressos e revistas, destacados logo abaixo:

- Optimizing tasks generation for children in the early stages of literacy teaching: a study using bio-inspired metaheuristics. *Soft Computing*, 1–14, 2018.
- Um relato de experiência de capacitação para a criação de jogos sérios em cursos de computação com foco em projetos reais. *RENOTE – Revista de Novas Tecnologias na Educação*, 14(1), 2016.
- Jogos digitais e aplicações web como forma de engajamento no ensino da leitura utilizando tarefas adaptativas matching-to-sample. *IE Comunicaciones: revista iberoamericana de informática educativa. Informatica Educativa Comunicaciones*, 27(27), 2018.

- Engagement in digital games and web applications using adaptive matching-to-sample tasks in teaching reading. In *2017 International Symposium on Computers in Education (SIIE)* (pp. 1–6). IEEE. 2017.
- Um framework para a geração de repertórios de ensino individualizado baseado em dificuldade adaptativa. In *XLVIII SBPO Simpósio Brasileiro de Pesquisa Operacional*. Vitória. 2016.
- ABSR: An algorithm based on similarity rules assisted by bio-inspired algorithms for teaching context. In *XLVIII SBPO Simpósio Brasileiro de Pesquisa Operacional*. Vitória. 2016.

1.6. Estrutura da Tese

Além deste capítulo introdutório, a tese fundamentada neste documento é composta por seis capítulos, incluindo apêndices e anexos.

O **Capítulo 2** apresenta o domínio de aplicação e o referencial teórico buscando apresentar os conceitos da análise comportamental, do procedimento *Matching-to-Sample*, a noção de dificuldade de uma tarefa de ensino e as técnicas de computação bio-inspiradas.

O **Capítulo 3** apresenta a pesquisa de trabalhos correlatos envolvidos nesta tese. É efetuado uma análise sucinta de todos os trabalhos relacionados mostrando sua contribuição e o diferencial com a linha de pesquisa proposta nesta tese.

O **Capítulo 4** expõem a abordagem para a geração de tarefas de ensino adaptadas através de algoritmos bio-inspirados, que engloba a dificuldade adaptada de uma tarefa de ensino, o fluxo de dificuldade de ensino, a utilização dos algoritmos bio-inspirados e os softwares de aprendizados implementados.

O **Capítulo 5** exhibe as atividades e os experimentos para validação desta tese assim como os seus resultados.

E por fim, o **Capítulo 6** conclui este trabalho discutindo os resultados produzidos, as contribuições reveladas e indica trabalhos futuros neste ramo de conhecimento.

2. Referencial Teórico

2.1. Considerações iniciais do capítulo

A utilização de sistemas de aprendizagem fornece oportunidades para novos caminhos na aplicação da Inteligência Artificial (IA). As técnicas desenvolvidas no campo de IA para a Educação (IAED) auxiliam na resolução de problemas de diversos tópicos de ensino, como: interação estudante-tutor, modelagem de domínio de ensino, avaliação integrada, recomendação de tutoriais, mineração de dados na educação, jogos sérios motivacionais, *feedback* personalizado e agentes pedagógicos. A IAED se tornou parte integrante e fundamental para auxiliar no processo de aprendizagem, proporcionando métodos de estimulação no desenvolvimento cognitivo do aprendiz, com auxílio e criação de estratégias de entendimento para a solução de diversos problemas (Blanchard, 2014). Trabalhos sobre a modelagem cognitiva do aluno com o objetivo de inferir o aprendizado, tem sido fundamental para a formalização de modelos cognitivos. Nestes modelos cognitivos, muitas vezes a experiência de especialistas é mapeada e repassada para o aluno, principalmente quando o objetivo é a aquisição de habilidades para resolver problemas complexos (C.-M. Chen & Duh, 2008; Nešic, Gašević, Jazayeri, & Landoni, 2011). Mais recentemente foram discutidas etapas intermediárias na interseção entre psicologia comportamental e computação, como a realização do diagnóstico prévio do aluno, suas habilidades e a mensuração da dificuldade das atividades de ensino (Bolkan, 2016; L. X. Chen & Sun, 2016; Hwang & Wang, 2016).

Nos próximos subcapítulos, são apresentados os conceitos da análise comportamental e modelos matemáticos que serviram como base para a concepção da dificuldade de uma tarefa de ensino adotadas neste trabalho. Também serão expostas as teorias das técnicas de inteligência computacional conhecidas como algoritmos bio-inspirados, utilizadas como o núcleo de implementação desta tese.

2.2. Análise comportamental e procedimento *Matching-to-Sample* aplicados no processo de ensino da leitura

A análise comportamental é um campo da psicologia que observa o comportamento de um indivíduo no processo de aprendizagem, com o objetivo de auxiliar no delineamento de características importantes para favorecer o ensino (Todorov, 1982).

Agregado aos vários conceitos da análise comportamental deve-se denotar o conceito de **condicionamento operante**, que segundo Mizukami (1986), é um: *“comportamento voluntário e abrange uma quantidade muito maior da atividade humana - desde os comportamentos do bebê de balbuciar, agarrar objetos, olhar os enfeites do berço, até os comportamentos mais sofisticados de um adulto”*.

O condicionamento operante utiliza, dentro de seu escopo, a **teoria da equivalência de estímulos**. Essa teoria estabelece critérios de relações simbólicas conhecidas como **estímulos** (Bortoloti & Rose, 2007). Pode-se considerar os estímulos como símbolos (texto, imagens, sons, vídeos, animações, etc.) e essas representações simbólicas de estímulos são chamadas de **classes de estímulos**. As classes de estímulos podem ser relacionadas a fim de estabelecer a associação entre os estímulos. Neste contexto, os procedimentos comportamentais de associação entre estímulos são considerados métodos de ensino que podem ser adequados às necessidades de um indivíduo (Cavalcante, 1997). Por exemplo: a figura de uma “Casa” tem relação com a palavra escrita em formato textual “CASA”, que por sua vez tem relação com o som da palavra ditada por uma pessoa.

Da mesma forma, os analistas comportamentais trabalham com a interação entre as classes de estímulos e classes de respostas. Existe também as **classes das respostas**, que são os retornos que um indivíduo produz após a apresentação desses estímulos. Por exemplo: ao visualizar o estímulo “Casa” em forma de figura, som ou texto, o indivíduo atua com uma resposta, que pode ser: falar, sinalizar, escrever, digitar, sequenciar ou até mesmo não realizar nenhuma ação.

Após a apresentação desses estímulos, e aliado a um *feedback* adequado, o indivíduo pode manter a associação de estímulos em um comportamento de aprendizado. Para isso é utilizado o **reforçamento** ou **reforço**, que é um conceito importante na análise comportamental aplicada ao ensino. Pode-se entender por reforçamento qualquer operação que altere a chance de uma resposta ocorrer no futuro (Rodrigo, 2009). Entre os tipos de reforços existentes pode-se citar:

- reforço positivo: também chamada de uma consequência positiva, é uma operação em que um evento produzido por uma resposta aumenta a probabilidade desta resposta ocorrer no futuro. Pode ser representado por um estímulo de incentivo, identificando que o indivíduo está no caminho certo.

- reforço negativo: também chamada de consequência negativa, é marcado pela eliminação de um evento. O reforçamento negativo, representado por um estímulo negativo ou aversivo, é uma operação em que uma resposta tem sua probabilidade de ocorrência aumentada pela eliminação de um estímulo.
- reforço neutro: significa uma consequência neutra ou ausência de estímulos de reforço.

O reforçamento dentro do contexto de ensino-aprendizagem pode também ser utilizado na **discriminação condicional** (Debert 2016), onde:

“uma situação de discriminação condicional não se estabelece uma relação constante entre um estímulo antecedente e uma resposta; esta relação muda de acordo com os contextos nos quais este estímulo aparece. Em linhas gerais, para se obter uma relação condicional, deve-se reforçar determinada resposta na presença de um estímulo específico apenas se um outro estímulo estiver presente. Apenas na presença desta combinação de dois estímulos, as respostas são seguidas de reforço. Combinações outras desses estímulos com outros estímulos não se configuram ocasiões diante das quais as respostas são seguidas de reforço, mesmo se estas combinações compartilharem um dos estímulos com as combinações diante das quais as respostas foram seguidas de reforço”.

Neste contexto, conforme Sidman & Tailby (2006), uma **tarefa de ensino** ou uma **tentativa de ensino** é um arranjo de classes de estímulos aliado a uma relação associativa dentro do contexto da análise comportamental. Diversas tarefas de ensino podem ser adicionadas e encadeadas em um **programa de ensino**. Conforme Juliano (2009), um programa de ensino é uma combinação de tarefas apresentadas a um indivíduo, e com base nas respostas das tarefas realizar a organização de tarefas em ordem crescente de dificuldade, visando aumentar gradualmente esse repertório de tarefas via **reforçamento positivo**.

A modelagem de uma tarefa segundo os conceitos da análise comportamental pode ser estruturada envolvendo estímulos, respostas do indivíduo, reforços positivos em caso de progresso, reforços negativos em caso de falhas, ou reforços neutros. Assim, pode-se considerar que um programa de ensino é um conjunto de blocos de tarefas, onde cada bloco busca ensinar algum conteúdo.

Estes programas de ensino utilizam meios de apresentar uma tarefa de ensino. Um dos meios bem utilizados no âmbito da análise comportamental é o procedimento de emparelhamento ao modelo de referência ou *Matching-To-Sample* (Mackay, 1985; Reis,

Souza, & Rose, 2009). O procedimento MTS é utilizado tanto para treino (ensino) quanto para o teste (avaliação) da relação de leitura.

Neste procedimento, quando há a presença de um estímulo modelo, o aluno deve selecionar uma opção. A seleção da opção correspondente ao modelo gera um reforço positivo para as tarefas na fase de treino de palavras. Esse reforço estabelece a relação entre a imagem modelo e a palavra selecionada. A configuração de uma tarefa de MTS garante que o estímulo modelo possa ser claramente diferenciado dos estímulos alternativos.

Como exemplo prático destes conceitos, destaca-se o programa de ensino “Aprendendo a Ler e a Escrever em Pequenos Passos” (ALEPP) (J. Rose et al., 2012; T. Smith, 2001) que busca auxiliar crianças que possuem dificuldade em leitura e escrita. O ALEPP pode servir de auxílio para crianças que não possuem desejo em aprender por intermédio de um tutor ou professor. Isto geralmente ocorre com crianças com déficit de aprendizado, crianças com autismo ou crianças com síndrome de *down*. O formato de atividades para o aprendizado, como procedimentos derivados da associação por equivalência de estímulos podem criar clareza para a criança. Assim o ALEPP trouxe uma alternativa de ensino para esse público alvo maximizando o sucesso (T. Smith, 2001).

Este programa de ensino permite que a pessoa cumpra um fluxo de ensino conforme seu ritmo de aprendizado. As tarefas compõem blocos ou repertórios de tarefas de ensino que também são chamadas de unidades de ensino. O aprendiz pode repeti-las, caso tenha dificuldade em algum momento do seu progresso. O ALEPP estabelece uma relação entre palavras ditadas, áudios, imagens, palavras impressas e as classes de respostas de seleção através do procedimento MTS.

Inicialmente o ALEPP era conduzido e reproduzido por cartolinas, fotos e objetos. Atualmente alguns sistemas e softwares mantêm o programa de ensino informatizado (Orlando, 2009).

A Estrutura do ALEPP permite o uso de tarefas por manipulação de características de ensino, de modo que a dificuldade do que é exigido para o usuário pode ser modulada de forma manual. Esta modulação de dificuldade no programa de aprendizagem permite que o sistema de ensino se torne customizado. O programa é dividido em unidades de ensino, e de cada unidade de ensino é subdividida em sessões que contêm um conjunto de tarefas de ensino.

As tarefas de ensino são rotuladas de acordo com as relações entre as modalidades da palavra, que são organizadas com modelos e escolhas. Por exemplo: uma relação texto-som pode ocorrer em uma tarefa, onde há um som da palavra ditada (modelo) e algumas palavras escritas (escolhas).

Para facilitar a nomenclatura dos tipos de tarefas, utilizam-se letras para identificar cada formato de estímulo. A letra “A” é usada para representar um estímulo sonoro, a letra “B” é utilizada para representar uma imagem, e “C” é um texto impresso. Assim, a relação de “som-texto” é chamada de “tarefa do tipo AC”, onde a primeira letra indica sempre o modelo e a segunda letra as opções.

A Tabela 2.2 ilustra como os tipos de tarefas estão relacionados. A tarefa do tipo “AB” exige que o usuário visualize algumas figuras e selecione aquela que está relacionado com a palavra ditada “bolo”. O próximo tipo de tarefas “AC” pede ao aluno para selecionar uma palavra escrita, por exemplo, “tatu” quando a mesma palavra é ditada. A tarefa “BC” pede ao aluno para encontrar uma palavra escrita, “apito”, utilizando uma imagem de um apito. Finalmente, o tipo de tarefa “CB” exige que o aluno selecione a imagem correspondente da palavra escrita, “cavalo”.

Tabela 2.2 – Exemplos de tipos de relações de tarefas do procedimento MTS.

Tipo de Relação entre tarefas MTS	Ilustração da tarefa	Função do Aluno
AB	 <bolo>   	Diante da instrução falada “Aponte bolo”, o aluno deverá selecionar a figura do ‘bolo’.
AC	 <tatu> cavalo tatu bico	Diante da instrução falada “Aponte tatu”, o aluno deverá selecionar a palavra escrita «tatu».
BC	 vaca apito tatu	Diante da figura de um apito, o aluno deverá selecionar a palavra escrita «apito».
CB	cavalo   	Diante da palavra escrita «cavalo», o aluno deverá selecionar a figura «cavalo».
BB	   	Diante da figura «mala», o aluno deverá selecionar a figura «mala».
CC	bico cavalo apito bico	Diante da palavra escrita «bico», o aluno deverá selecionar a palavra «bico».

Outro procedimento incluso neste programa de ensino é o procedimento CRMTS (*Constructed-Response MTS*) ou construção da resposta (S. de Souza, Goyos, Silveires, &

Saunders, 2007). Ele consiste na seleção do estímulo de forma que se construa o modelo, parte por parte, até a estrutura se tornar equivalente ao modelo de referência. Esse procedimento é muito utilizado para montar palavras, letra por letra ou sílaba por sílaba, até apresentar a palavra totalmente construída e equivalente com a palavra de referência. Pode ser feita uma analogia com o popular jogo da “forca” que ensina a soletração.

Para este tipo específico de procedimento, a representação utilizada é representada pela letra “E”. Alguns tipos de relações entre tarefas do procedimento CRMTS se encontram na Tabela 2.3.

Tabela 2.3 – Três exemplos de diferentes relações entre tarefas do tipo CRMTS.

Tipo de Relação entre tarefas CRMTS	Ilustração da tarefa	Função do Aluno
AE	 <cavalo> <hr/> ca bo va ca lo va lo	Diante da instrução falada “Escreva cavalo”, o aluno deverá compor a palavra «cavalo», escolhendo as sílabas na ordem correta.
BE	 <hr/> ta ma mu ta te le fi to	Diante da instrução falada “Que figura é essa?”, o aluno deverá compor a palavra «tomate», escolhendo as sílabas na ordem correta.
CE	vaca <hr/> va ma co la va bi ca lo ca	Diante da instrução falada “Que palavra é essa?”, o aluno deverá compor a palavra «vaca», escolhendo as sílabas na ordem correta.

O programa ALEPP já foi utilizado largamente em estudos comportamentais, contudo algumas lacunas são identificadas no programa (D. G. de Souza et al., 2009; Orlando, 2009). A mensuração das dificuldades das tarefas de ensino, o reconhecimento das habilidades do aluno e a geração automática de tarefas de ensino são algumas das lacunas que poderiam ser estudadas para a integração em um novo ambiente de

aprendizado com sistemas inteligentes e possibilidade de aumento do número de tarefas e alunos sem perda de desempenho computacional.

2.3. Noções sobre a dificuldade de uma tarefa de ensino

Alguns estudos indicam que a dificuldade no processo de ensino pode ser de fundamental importância para manter o aluno focado e com bom desempenho. Conteúdos de ensino, materiais, cursos, itens e tarefas quando muito fáceis ou muito difíceis podem desestimular ou interferir no processo de ensino (De-Marcos et al., 2014). Este aspecto sugere uma relação entre motivação, recompensa e desafios em sistemas de aprendizagem interativos (Ronimus, Kujala, Tolvanen, & Lyytinen, 2014; Sampayo-Vargas, Cope, He, & Byrne, 2013).

Alguns pesquisadores também apontam uma correlação entre a dificuldade e as habilidades da pessoa em manter foco em sua atividade (L. X. Chen & Sun, 2016; Haworth & Sedig, 2011). Segundo Conejo *et al.* (2014), uma tarefa individualizada deve ter ajustes na dificuldade para garantir que o engajamento do aluno seja mantido. Em consequência, ao buscar a efetividade do processo de aprendizagem para cada aluno, o grau de dificuldade de cada tarefa pode ser personalizado de acordo com a competência individual. Assim, no processo de aprendizado de um aluno, o tutor de ensino possui o controle suficiente para ajustar um fluxo de ensino em sistemas de aprendizado que garantam o engajamento.

Foi recentemente observado que muitos sistemas adaptativos podem ser agregados a um ambiente computacional interativo para proporcionar melhorias no processo de ensino e aprendizagem de vários ramos do conhecimento. Exemplos desses ambientes variam de cursos on-line, sistemas baseados na Web, plataformas de redes sociais cooperativas e competitivas, até jogos educacionais que ajudam na satisfação do usuário e no crescimento do engajamento (Cabada et al., 2011; C.-M. Chen, 2008; C.-M. Chen & Duh, 2008; C. H. Chen & Law, 2015; L. X. Chen & Sun, 2016; Hwang & Wang, 2016; Klinkenberg, Straatemeier, & van der Maas, 2011; Nešic et al., 2011; G. G. Smith et al., 2013; Verdú, Regueras, De Castro, et al., 2012).

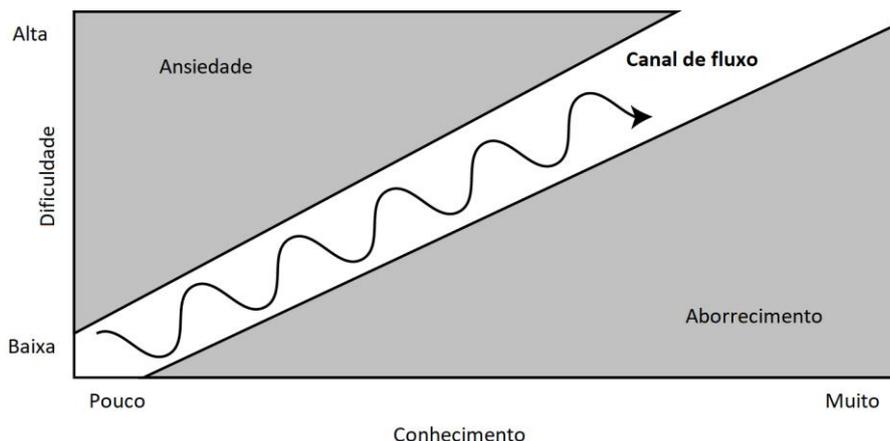
Nesse contexto, os sistemas adaptativos podem ajudar os educadores que desejam criar automaticamente uma tarefa individual para cada aluno, em vez de criar tarefas manualmente. Esses sistemas são implementados porque a criação de cada tarefa

manualmente é um método subjetivo que geralmente depende de um especialista para gerar cada tarefa com base em seu conhecimento e na identificação das habilidades do aluno (Hanus & Fox, 2014; Hwang & Wang, 2016; Jegatha Deborah, Baskaran, & Kannan, 2012). Assim, esses sistemas têm o meio computacional para recomendar um conjunto apropriado de tarefas, atividades, cursos, perguntas e materiais para a educação de um indivíduo (Dung & Florea, 2012).

Por exemplo, Marques *et al.* (2013) usou o programa de ensino ALEPP em um jogo digital com uma trama lúdica e uma transição progressiva de tarefas de aprendizagem com base no desempenho do aluno. O autor confirmou que o uso de multimídia interativa despertou os interesses do aluno com características que podem ser determinantes para manter o envolvimento e a motivação do jogador (Battaiola, Martins, Lívia, & Barbosa, 2008; Cezarotto & Battaiola, 2016; Kim et al., 2015; McGonigal, 2011). Contudo, ele não utilizou nenhum sistema computacional e adaptativo para ajustar a dificuldade das tarefas de ensino. Além disso, a grande maioria dos estudos não utilizam as tarefas MTS como forma de promover o aprendizado, mas possuem bons resultados usando aplicações motivacionais e convencionais (L. X. Chen & Sun, 2016; Hwang & Wang, 2016; G. G. Smith et al., 2013).

No contexto relacionado com o engajamento em uma atividade, pesquisadores descrevem a importância do canal de fluxo, ou *flow channel*, que é definido como um estado mental em que uma pessoa está totalmente imersa em uma atividade e com foco em sua realização (Haworth & Sedig, 2011). O canal de fluxo correlaciona a dificuldade da atividade com as habilidades da pessoa, a fim de manter essas variáveis equilibradas (L. X. Chen & Sun, 2016). Tratando-se do ensino-aprendizado, o canal de fluxo pode ser um meio para definir um caminho de aprendizagem. Nesta tese, por considerar que trataremos de valores de dificuldade, chamamos o canal de fluxo de fluxo de dificuldade. A Figura 2.1 mostra um gráfico que representa o canal de fluxo. Nele é observado que quanto menor o conhecimento e maior a dificuldade, maior a ansiedade ou cansaço mental na realização de uma atividade. Já quanto maior o conhecimento e menor a dificuldade, maior o aborrecimento ou a frustração em realizar uma determinada atividade.

Figura 2.1 – Exemplo de um Canal de fluxo



Fonte: Adaptado de "*Understanding the Flow Channel in Game Design*" (2015).

Levando em consideração somente uma tarefa, existe diversas maneiras de estimar sua dificuldade. A definição pode ser baseada somente em um indivíduo ou em vários, pode ser definida por meio estatístico através de uma amostra de dados, por modelos matemáticos baseado em equações, por algoritmos de regras previamente definidas por especialistas, ou por uma mesclarem híbrida destes métodos. Esta definição depende do contexto em que o domínio de aplicação se encontra (Verdú et al., 2012).

Nota-se modelos de estimação da dificuldade presentes na teoria de resposta ao item (TRI), que considera o item (tarefa) como uma unidade de análise. Essa teoria procura representar a probabilidade de um indivíduo responder corretamente uma questão baseado em seu conhecimento individual (Andrade, Laros, & Gouveia, 2010) e é muito utilizada em Testes Adaptativos Computadorizados (TAC).

Diferentes modelos do TRI podem ser selecionados de acordo com certas características dos itens o que inclui se os itens possuem duas ou mais categorias de respostas, um número de habilidades que podem ser mensuradas para um indivíduo (pessoa participante do processo de avaliação) e acerto ao acaso (quando uma pessoa acerta um item acidentalmente). Nessa metodologia de avaliação, o conhecimento do indivíduo é estimado e a seleção do item que será selecionada é feita com base nessa estimativa (Sharkness, 2014). Nesta tese, uma tarefa de ensino é similar a um item do TRI.

Ricarte (2013) faz uma revisão destes modelos onde o modelo de 1 parâmetro e o modelo de 3 parâmetros são os mais utilizados. O primeiro modelo, apresentado na

Equação 2.2, é o modelo de 1 parâmetro, também conhecido como modelo Rasch (Shankness, 2014).

$$P(U_{ij} = 1 | \theta_j; b_i) = \frac{1}{1 + e^{-1,7(\theta_j - b_i)}} \quad (2.2)$$

Onde:

j : indexa os indivíduos, ou seja, identificador da pessoa que fez os testes;

i : indexa os itens;

U_{ij} : é uma variável dicotômica que assume os valores 1, quando o indivíduo j responde corretamente o item i , ou 0 quando o indivíduo j não responde corretamente o item i ;

θ_j : é valor do conhecimento do indivíduo j ;

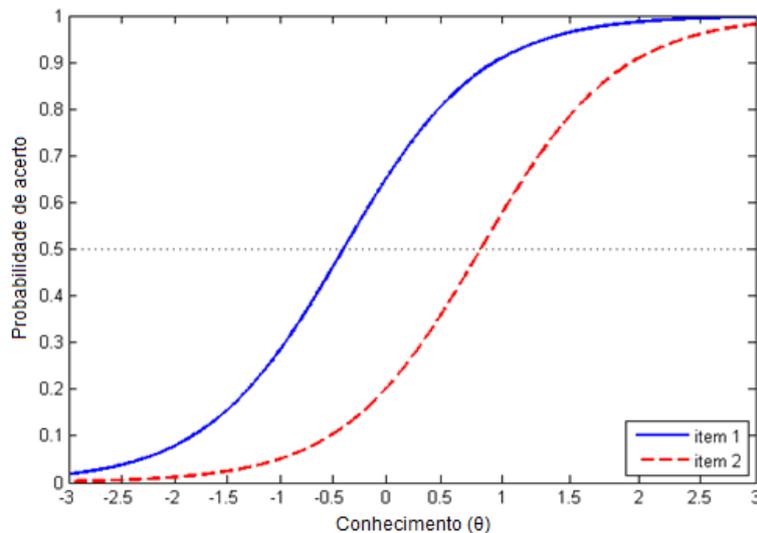
$P(U_{ij} = 1 | \theta_j; b_i)$: é a probabilidade de um indivíduo j com conhecimento θ responder corretamente o item i ;

e : é um número cujo valor é 2,718, base dos logaritmos neperianos;

b_i : é o parâmetro relativo à dificuldade do item i ;

Para a função do modelo de 1 parâmetro, pode-se gerar a Curva Característica do Item (CCI), que é utilizada para relacionar a probabilidade de resposta em função do conhecimento do aluno. A Figura 2.2 representa a CCI para o modelo de 1 parâmetro da Equação 2.2.

Figura 2.2 – Comportamento da curva característica do item para o modelo de 1 parâmetro com dois itens.



Neste exemplo, à medida que o conhecimento do indivíduo aumenta sobre o item 1, a probabilidade de acerto também aumenta; mas no item 2, a probabilidade de acerto aumenta mais tardiamente em relação ao item 1. Levando em consideração que esses dois itens são expostos para um mesmo aluno, pode-se considerar que o item 2 é mais difícil que o item 1, pois a probabilidade de acerto do item 2, demora a crescer em relação ao item 1.

Outro modelo logístico é o de 3 parâmetros (Baker & Al-Karni, 1991), que acrescenta parâmetros de análise para a discriminação do item e probabilidade de acerto ao acaso. A expressão matemática é exibida na Equação 2.3.

$$P(U_{ij} = 1|\theta_j; a_i; b_i; c_i) = c_i + \frac{1 - c_i}{1 + e^{-1,7a_i(\theta_j - b_i)}} \quad (2.3)$$

Onde se acrescenta à Equação 2.2 as seguintes variáveis:

a_i : é o parâmetro de inclinação do item, também chamado de parâmetro de discriminação;

c_i : é o parâmetro de probabilidade de acerto ao acaso do item i ; reflete as chances de um examinando de proficiência muito baixa selecionar a opção de resposta correta.

Após a definição do modelo, os parâmetros podem ser estimados através de um banco de itens com os parâmetros (a_i , b_i , c_i) já definidos. Geralmente, para calibrar esse

banco de itens, um pré-teste é realizado com vários indivíduos sobre um subconjunto de itens do banco.

Nesta tese os modelos do TRI serviram de inspiração para a modelagem da equação de dificuldade adaptativa apresentada no Capítulo 4 deste documento.

2.4. Computação bio-inspirada

Muitas vezes a complexidade de cálculos para um problema NP-difícil (*Non-deterministic Polynomial time* - Tempo polinomial não determinístico) e a identificação de uma solução é computacionalmente custosa, inviável em tempo limitado (Panchal & Panchal, 2015). O escopo grande e dinâmico de soluções dificulta a busca de soluções ótimas ou sub-ótimas.

Neste contexto, a biologia se tornou fonte de inspiração para enfrentar problemas computacionais, onde alguns pesquisadores criaram algoritmos baseado em organismos biológicos ou fenômenos da natureza (Kar, 2016). Este ramo da ciência da computação chama-se computação bio-inspirada, que tem solucionado problemas que variam desde encontrar a melhor configuração de um grande número de possíveis layouts de circuitos de hardware até realizar a busca de um conjunto de equações para estimar a capacidade cognitiva de um indivíduo em uma aplicação de aprendizagem (Mitchell & Taylor, 1999).

O foco desta tese foi os algoritmos genéticos, algoritmos de otimização por enxame de partículas e algoritmo de otimização por colônia de formiga. Existem dezenas de modelos e tipos de algoritmos bio-inspirados na literatura, contudo escolhemos os algoritmos citados pois são classes de meta-heurísticas de otimização representativas com maior utilização na comunidade científica (Kar, 2016), além de sua estrutura de variáveis poderem ser adaptadas para os diferentes desafios de busca. Assim, o problema de geração de tarefas adaptadas é um problema de busca combinatorial com diversas soluções possíveis, logo a utilização de uma meta-heurística bio-inspirada se mostra adequada para resolução deste problema.

2.4.1. Algoritmos genéticos

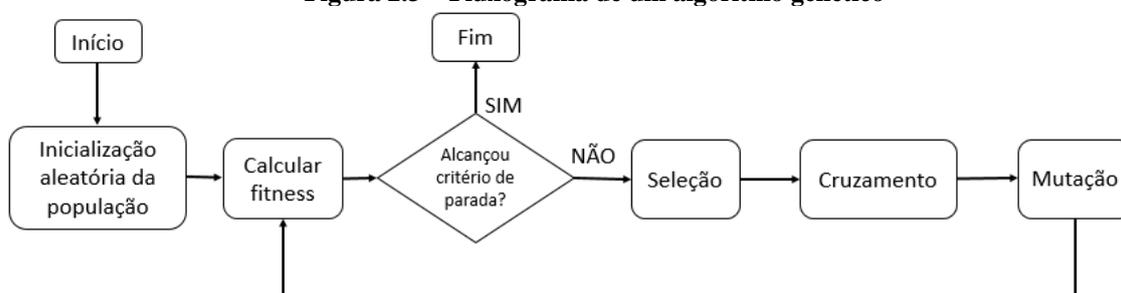
O algoritmo genético (*Genetic Algorithm - GA*) é um método de busca heurística adaptativa baseada na simulação da evolução das espécies, seleção natural e genética natural introduzido por John Holland no início da década de 1970 (Holland, 1975).

O GA busca encontrar soluções exatas ou aproximadas para otimização de busca e problemas. Suas vantagens são: capacidade de resolver uma grande quantidade de problemas que podem ser executadas em um microcomputador, eficiência de encontrar um ponto ótimo ou sub-ótimo de problemas com múltiplas soluções, possuir flexibilidade em se adequar a várias restrições e facilidade de adaptação quanto ao uso de dados discretos ou contínuos (Sun, Chen, Tsai, & Cheng, 2008).

Os algoritmos genéticos têm sido usados em diversas aplicações para encontrar as melhores soluções de problemas complexos em vários domínios, entre eles para sistemas educacionais (Huang, Huang, & Chen, 2007; Verdú et al., 2012).

O GA é uma simulação da seleção natural e evolução genética de melhores cromossomos dentro de sucessivas gerações. Os cromossomos com altos valores de aptidão possuem alta probabilidade de serem selecionados e cromossomos da nova geração podem ter maior valor médio de aptidão do que os cromossomos da geração anterior. O processo de evolução é repetido até que a condição de parada seja satisfeita. A Figura 2.3 mostra um fluxograma do algoritmo genético.

Figura 2.3 – Fluxograma de um algoritmo genético



Em contraste com os métodos de busca local, os algoritmos genéticos são considerados métodos de busca global por causa da manipulação de várias possíveis soluções ao mesmo tempo, e baseiam-se em um conjunto de cálculos independentes controlados por uma estratégia probabilística. As várias possíveis solução são geradas aleatoriamente de forma espalhada no espaço de busca e forma a população inicial do

AG, sobre a qual serão aplicados os cálculos de fitness, o procedimento de seleção e os operadores de cruzamento e mutação.

O tamanho da população inicial de cromossomos pode ser determinado de acordo com a complexidade do problema. Uma grande população reduz a velocidade de busca do algoritmo genético, mas pode aumentar a probabilidade de encontrar uma solução de alta qualidade.

Cada indivíduo tem uma cadeia de cromossomo que codifica suas características de dados. Logo, um cromossomo é uma sequência de alelos que representam um quantum de informações, tais como um bit, um dígito, uma letra. É uma representação de dados alternativa que requer codificação e decodificação, a fim de trocar soluções com o espaço de soluções possíveis.

A função objetivo gera um índice de aptidão (valor de *fitness*), que é aplicado para avaliar a qualidade dos cromossomos durante os ciclos do algoritmo. Após o início do ciclo de gerações, é realizada a operação de seleção onde os cromossomos de melhor valor da função objetivo têm uma probabilidade relativamente maior de gerar filhos para a próxima geração. Esta operação tem como objetivo selecionar boas soluções para atingir a meta de evolução genética. Existem diversas técnicas para a seleção de cromossomos, dentre as mais conhecidas estão a seleção por roleta e seleção por torneio (Abuiziah & Shakarneh, 2014; Butz, Sastry, & Goldberg, 2003).

Uma forma particular de operação de seleção é o elitismo, que ocorre como forma de selecionar alguns cromossomos com melhor avaliação de *fitness* e preservá-los para a próxima geração.

Após as operações de seleção, entra em cena a operação de cruzamento, que realiza a combinação de dois indivíduos. Seu objetivo é inserir as melhores características de ambos os pais aos filhos que serão gerados. Existem três principais técnicas de cruzamento:

- 1) Cruzamento de um ponto: é a mais simples de todas as técnicas. Uma posição aleatória, no cromossomo, é selecionada para o cruzamento e parte das características dos pais é copiada para os seus filhos, conforme a Figura 2.4.

Figura 2.4 – Esquema de cruzamento de um ponto

PAI 1	AAA AAAA
PAI 2	BBB BBBB
FILHO 1	AAA BBBB
FILHO 2	BBB AAAA

- 2) Cruzamento de dois pontos: neste método seleciona-se dois pontos aleatórios, no cromossomo, então, o filho 1 recebe parte intermediária do pai 2 e o restante é do pai 1. O mesmo processo ocorre para o filho 2, conforme a Figura 2.5.

Figura 2.5 - Esquema de cruzamento de dois pontos

PAI 1	AA AAA AA
PAI 2	BB BBB BB
FILHO 1	AA BBB AA
FILHO 2	BB AAA BB

- 3) Cruzamento uniforme: neste, o cruzamento é feito em vários pontos, mas, estes são uniformemente distribuídos pelo cromossomo que pode ter ordenação par ou ímpar, conforme Figura 2.6.

Figura 2.6 - Esquema de cruzamento uniforme

PAI 1	A A A A A A A
PAI 2	B B B B B B B
FILHO 1	A B A B A B A
FILHO 2	B A B A B A B

A operação de mutação geralmente é executada após o cruzamento de cromossomos. Essa operação pode gerar alguns novos indivíduos que não podem ser produzidos pelas operações de cruzamento.

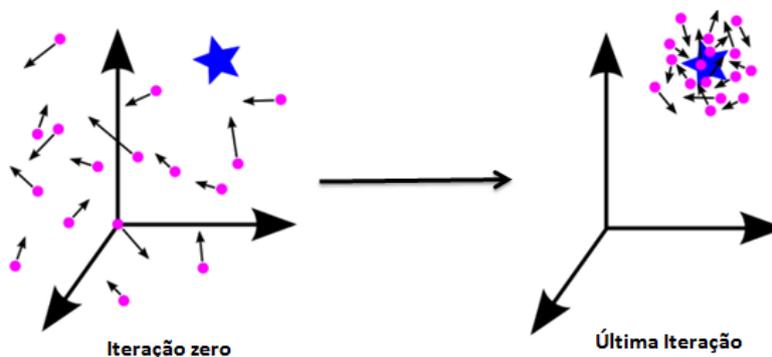
O algoritmo realiza os passos de cálculo do *fitness*, seleção, cruzamento e mutação repetidas vezes até atingir um critério de parada determinado pelo projetista do algoritmo evolucionário. Esta técnica pode ser facilmente adaptado para o problema de geração de tarefas utilizando uma codificação de variáveis discretas que represente elementos das tarefas de ensino para a leitura.

2.4.2. Algoritmo de otimização por enxame de partículas

O algoritmo de otimização por enxame de partícula ou PSO (*Particle Swarm Optimization*), foi formulado por Kennedy (1995) e inspirado na revoada de pássaros ou cardumes de peixes em busca de comida. Os animais, especialmente as aves e peixes, se locomovem em grupos sem se colidirem. Cada membro segue o seu grupo, ajustam suas posições e velocidade usando as informações do grupo, para reduzir o esforço do indivíduo em busca de comida e abrigo.

O PSO é uma técnica desenvolvida para a otimização de funções não diferenciáveis multimodais, não lineares e contínuas que possui boa eficiência computacional e possui poucos parâmetros para ajuste (Calcada, Rosa, Duarte, & Lopes, 2010). No PSO, uma partícula representa um indivíduo e cada partícula tem um valor de *fitness* para estimar a proximidade da solução ótima. As partículas do enxame estão dispersas no espaço até convergirem para a solução ideal. Uma nova velocidade e as novas posições das partículas são calculadas para cada iteração do algoritmo, ajustando sua posição no espaço de busca de acordo com sua própria experiência anterior e a partir de seus vizinhos (Jiao, Liu, & Liu, 2008), como mostra o exemplo da Figura 2.7.

Figura 2.7 - Exemplo de um enxame de partículas.



Fonte: Adaptado de Machado (2016).

O processo de execução do PSO começa com a inicialização randômica das partículas e seus parâmetros. Em seguida é realizado o processo de movimentação das partículas baseado em cálculos de parâmetros de velocidade e posições da partícula (Equação 2.4).

$$V_{i+1} = wV_i + r_1c_1(M_i - X_i) + r_2c_2(Mg - X_i) \quad (2.4)$$

Onde:

X_i e V_i : são a posição atual e velocidade atual da partícula;

w : é o parâmetro de inércia da partícula;

c_1 : é o parâmetro cognitivo da partícula;

c_2 : é o parâmetro social da partícula;

r_1 e r_2 : são as variáveis aleatórias binárias com valor 0 ou 1;

M_i : é a melhor posição local já encontrada pela partícula;

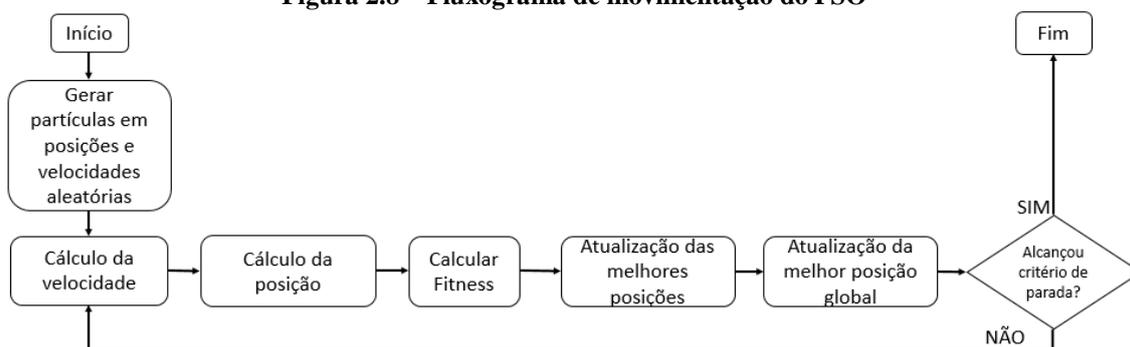
Mg : é a melhor posição global já encontrada pelo enxame.

O cálculo da nova posição da partícula (Equação 2.5) ocorre com a soma da velocidade atual com a posição atual.

$$X_{i+1} = V_i + X_i \quad (2.5)$$

A melhor posição das partículas é definida a partir do cálculo da função objetivo, criando a melhor posição local para cada partícula. Dentre as melhores posições de partículas, a melhor posição é armazenada atribuindo o nome de melhor posição global. O processo segue iterativamente até atingir um critério de parada. A Figura 2.8 mostra o fluxograma de movimentação e atualização dos elementos do enxame.

Figura 2.8 – Fluxograma de movimentação do PSO

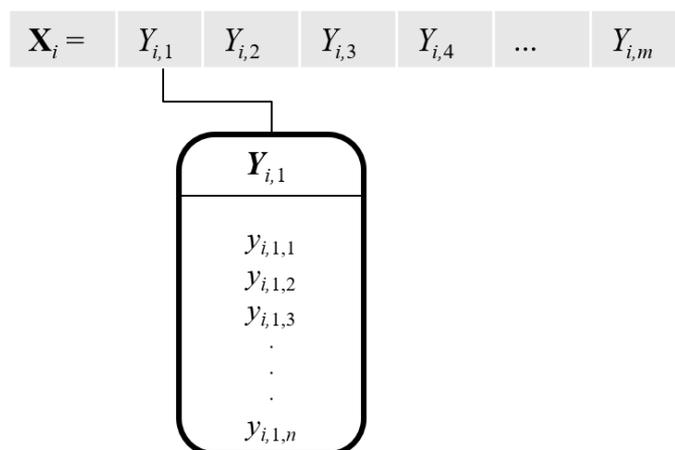


O PSO foi originalmente projetado para funcionar bem com dados contínuos. No entanto, ao usar soluções em um formato de dados discreto ou categórico, são necessárias algumas adaptações. O PSO inteiro e categórico (*Integer and Categorical Particle Swarm Optimization - ICPSO*) faz uso de valores discretos em sua estrutura de dados (Strasser, Goodman, Sheppard, & Butcher, 2016), e desta forma se mostra apto a ser utilizado em problemas onde as variáveis são múltiplas *strings*. Nesta tese, uma solução é formada por uma lista de tarefas que computacionalmente são vetores com

valores textuais, ou seja, valores discretos. Por esta razão o ICPSO se mostra adequado para ser utilizado neste trabalho.

No ICPSO, a posição da partícula é um conjunto de distribuições de probabilidades, e uma partícula é representada como $\mathbf{X}_i = [Y_{i,1}, Y_{i,2}, \dots, Y_{i,j}, \dots, Y_{i,m}]$, onde: \mathbf{X}_i é a posição da partícula i , $Y_{i,j}$ é a distribuição de probabilidade j da partícula i , com $1 \leq j \leq m$. A distribuição de probabilidade $Y_{i,j}$ é representada como $Y_{i,j} = [y_{i,j,1}, y_{i,j,2}, \dots, y_{i,j,k}, \dots, y_{i,j,n}]$, onde: $y_{i,j,k}$ é a probabilidade da variável atribuída para o k elemento, com $1 \leq k \leq n$; e n é o número de elementos possíveis para $Y_{i,j}$. A Figura 2.9 mostra uma representação da estrutura de partícula no ICPSO.

Figura 2.9 – Representação da estrutura da posição de uma solução no ICPSO.



A variável de velocidade neste algoritmo é estruturada similarmente a variável de posição da partícula. O ICPSO mantém as equações e procedimentos originais do PSO para variáveis contínuas, mas isso tem grande chance de os resultados das operações caírem fora do intervalo entre 0 e 1; por isso a cada iteração do algoritmo as variáveis são normalizadas, mantendo a validade da distribuição probabilística (soma igual a 1). Assim, o valor de *fitness* é calculado por uma amostragem através de um algoritmo de roleta, e segundo Strasser *et al.* (2016), este algoritmo alcança bons e estáveis resultados em comparação com outras versões de PSOs.

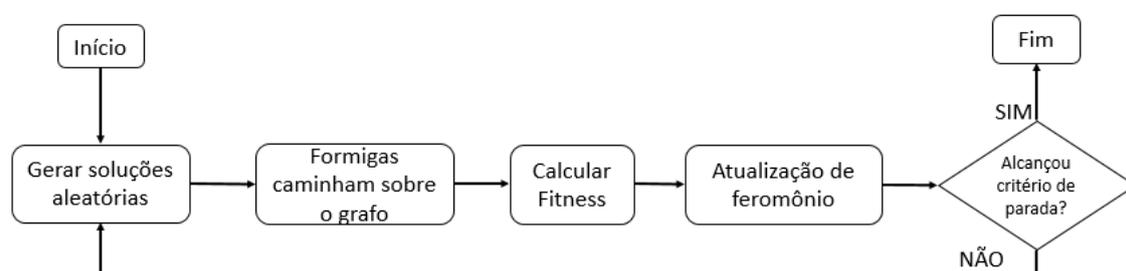
2.4.3. Algoritmo de otimização por colônia de formigas

O algoritmo de otimização por colônia de formigas (*Ant Colony Optimization - ACO*) é um algoritmo inspirado na natureza e foi apresentado na década de 1990 por Dorigo, Maniezzo & Colorni (1996). Como em uma colônia de formigas, as formigas

procuram alimento, e durante esse processo, depositam uma quantidade de feromônio no caminho percorrido. A maior quantidade de feromônio depositado em determinado caminho indica um caminho ótimo, por onde as outras formigas da colônia devem andar na busca por comida. O feromônio depositado em determinado caminho vai perdendo intensidade ao longo do tempo, então se outras formigas não passarem por esse caminho, o mesmo irá desaparecer do conjunto de possíveis rotas para as formigas.

Inicialmente, o algoritmo ACO foi usado efetivamente para problemas de busca de melhor caminho. No entanto, esse algoritmo pode ser projetado para outros problemas de otimização combinatória com algumas adaptações (Dorigo & Blum, 2005; Krynicki et al., 2016). O ACO também pode ser usado para dados contínuos e discretos e o processo de execução está representado da Figura 2.10.

Figura 2.10 – Fluxograma de execução de um algoritmo de ACO.

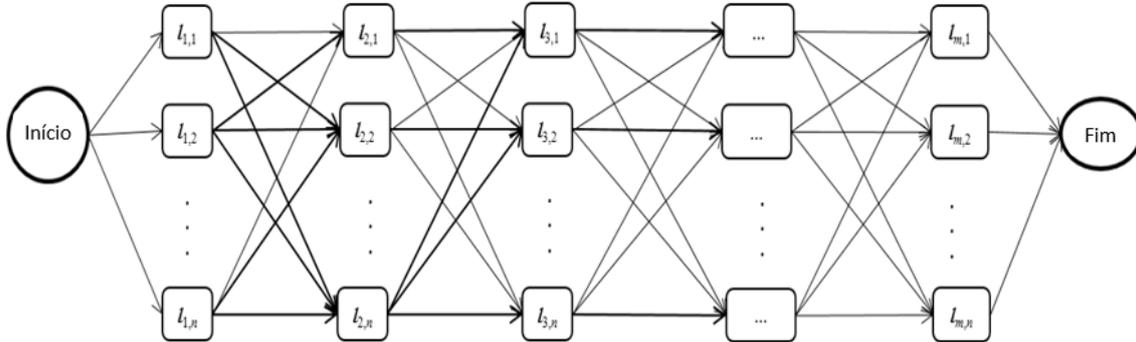


No ACO, um grafo é um dado de entrada onde os nós do grafo representam as “cidades” por onde as formigas podem caminhar. Um caminho completo por onde passou uma formiga representa uma solução do problema. As arestas entre os nós do grafo possuem um valor de probabilidade, que é a possibilidade de uma formiga escolher este caminho. A probabilidade das arestas é calculada através de um valor chamado de feromônio, que é depositado por cada formiga ao percorrer por este caminho. Este valor é atualizado com base no valor de *fitness* a cada iteração do algoritmo.

Assim, antes de usar o ACO é necessário projetar os caminhos pelos quais as formigas irão caminhar. Um exemplo de grafo é exibido na Figura 2.11, onde os nós são chamados l e as arestas são os caminhos que uma formiga pode percorrer. Nota-se que neste exemplo as arestas são setas unidirecionais, partindo de um nó e alcançando outro nó a sua frente. Neste exemplo adota-se a representação da aresta dos nós como $[l_{(x,z)}, l_{(x',z')}]$ indicando o nó de partida ($l_{(x,z)}$), e o próximo nó de chegada da aresta

$(l_{(x',z')})$, onde: x, z são os índices de partida da aresta; x', z' são índices de chegada da aresta; m, n são os índices dos elementos finais.

Figura 2.11 – Exemplo da estrutura de um grafo para o ACO.



A opção de qual aresta a formiga vai escolher pode ser feita com um algoritmo de roleta, e a equação que mostra o valor de probabilidade é exibido na Equação 2.6.

$$q_{[l_{(x,z)}, l_{(x',z')}]^k} = \frac{\tau_{[l_{(x,z)}, l_{(x',z')}]}}{\sum_{i=1}^n \tau_{[l_{(x,z)}, l_{(x',z')}]}} \quad (2.6)$$

Onde:

τ : é a quantidade de feromônio depositado na aresta entre os nós $l_{(x,z)}$ e $l_{(x',z')}$;

$q_{[l_{(x,z)}, l_{(x',z')}]^k}$: é a probabilidade de a formiga k escolher a aresta entre os nós $l_{(x,z)}$ e $l_{(x',z')}$.

Os feromônios referente a cada aresta são atualizados quando todas as formigas completaram seu caminho completo, onde as Equações 2.7 e 2.8 correspondem ao cálculo de atualização do feromônio.

$$\tau_{[l_{(x,z)}, l_{(x',z')}] = (1 - \rho)\tau_{[l_{(x,z)}, l_{(x',z')}] + \sum \Delta\tau_{[l_{(x,z)}, l_{(x',z')}]^k} \quad (2.7)$$

$$\Delta\tau_{[l_{(x,z)}, l_{(x',z')}]^k = \begin{cases} f, & \text{se a formiga } k \text{ percorreu a aresta entre os nós } l_{(x,z)} \text{ e } l_{(x',z')} \\ 0, & \text{caso contrário} \end{cases} \quad (2.8)$$

Onde:

ρ : é a taxa de evaporação do feromônio;

$\Delta\tau_{[l_{(x,z)}, l_{(x',z')}]^k$: a quantidade de feromônio depositado pela formiga k ;

f : é o *fitness* da solução.

O ACO é bem representativo no campo dos algoritmos bio-inspirados. A metodologia de agentes artificiais, que caminham por um espaço de busca, pode produzir bons resultados. Além disso, a estrutura de grafo permite uma flexibilidade para o uso de variáveis discretas, onde cada nó da solução pode ser uma *string*. Para o problema de geração de tarefas ensino, cada *string* faz parte da solução, e esta característica foi importante para selecionar o ACO como parte do presente estudo.

2.5.Considerações finais do capítulo

Este capítulo apresentou brevemente pontos relevantes desta tese iniciando por uma pequena análise sobre inteligência computacional aplicada a educação. Posteriormente foram descritos conceitos pertinentes sobre a análise do comportamento e o procedimento *Matching-to-sample*. Logo em seguida foram descritos, de forma sucinta, uma análise sobre a noção de dificuldade das tarefas de ensino e suas variações usadas em testes adaptativos computadorizados. Por fim, foi dado ênfase aos algoritmos de computação bio-inspirada.

Todos esses conceitos, técnicas, elementos e métodos serviram de inspiração para esta tese, onde alguns deles estão presentes de maneira total ou parcial. Os algoritmos bio-inspirados apresentados fizeram parte do estudo, sendo ajustados e testados diversas vezes para alcançar resultados satisfatórios. No Capítulo 4, referente ao sistema de geração de tarefas, são apresentados os caminhos práticos desta tese, mais precisamente a modelagem para a geração de tarefas de ensino adaptadas através de algoritmos bio-inspirados. Logo, os termos destacados no Capítulo 2 foram importantes e usados no Capítulo 4. Já no Capítulo 3, são apresentados as pesquisas e trabalhos correlatos a esta tese.

3. Trabalhos Correlatos

3.1. Considerações iniciais do capítulo

Neste capítulo são expostos os trabalhos relacionados com a presente tese. Esta pesquisa utilizou como referência, trabalhos que foram publicados em revistas, periódicos, anais de conferências e outros meios de pesquisa com importância científica entre os anos de 2010 e 2018, contudo alguns trabalhos mais antigos também são destacados devido à sua importância na área. Os repositórios e motores de busca de artigos utilizados foram: *IEEEExplore*, *ACM Digital Library*, *ScienceDirect*, *Springer*, *Google Scholar* e *ERIC website*.

A apreciação feita por essa tese procurou encontrar pontos de interseção com palavras-chave englobando os seguintes temas:

- Fatores de dificuldade de tarefas ou itens de ensino (*Keywords: difficulty, tasks, teaching tasks, teaching of reading*);
- Identificação do aprendizado de forma individualizada (*Keywords: learning, individualized learning, learning identification*);
- Geração, sugestão, recomendação ou seleção automática de tarefas (*Keywords: teaching tasks generation, teaching tasks suggestion, teaching tasks selection, teaching tasks recommendation*);
- Ensino da leitura via procedimentos de ensino para crianças (*Keywords: teaching procedures for children, behavioral procedures for reading, matching to sample procedure*).

Após a execução da busca de trabalhos científicos através das palavras chaves, a exclusão dos artigos que não se enquadravam a proposta desta tese foi feita analisando título e resumo. Os artigos selecionados foram armazenados em ferramentas de organização bibliográfica para análise futura. Após esta análise mais aprofundada, alguns artigos foram excluídos devido a não adequação a esta tese. Foram 123 trabalhos científicos selecionados que focaram no estado da arte, na área do domínio de aplicação ou trabalhos mais técnicos voltados utilizando práticas computacionais. Os 33 trabalhos correlatos mais relevantes estão presentes nesta tese.

Constatou-se que muitos dos trabalhos pesquisados não estavam voltados para o ensino utilizando procedimentos do contexto da análise comportamental ou para crianças em fase de aprendizado. Geralmente estes trabalhos estão fora do contexto da análise comportamental e estão incluídos em outras correntes pedagógicas e em sua maioria é aplicada para jovens e adultos. Outros trabalhos correlatos envolvem a análise comportamental sem a utilização de técnicas de inteligência computacional, onde a geração de tarefas de ensino foi feita de modo manual, baseado na experiência do especialista. E, apesar da carência de artigos para o termo “geração automática de tarefas de ensino para a leitura”, buscou-se analisar se as técnicas de inteligência computacional, os métodos estatísticos e as operações de customização manuais, poderiam ser adaptados a esta tese.

Com o objetivo de categorizar as pesquisas conforme o uso e finalidade, os trabalhos foram divididos em: Sistemas de ensino adaptáveis (sistemas customizados por especialistas); Sistemas de tutoria e aprendizado adaptativos (sistemas inteligentes de ensino-aprendizado); e Sistemas de testes adaptativos (sistemas para avaliação do aprendizado).

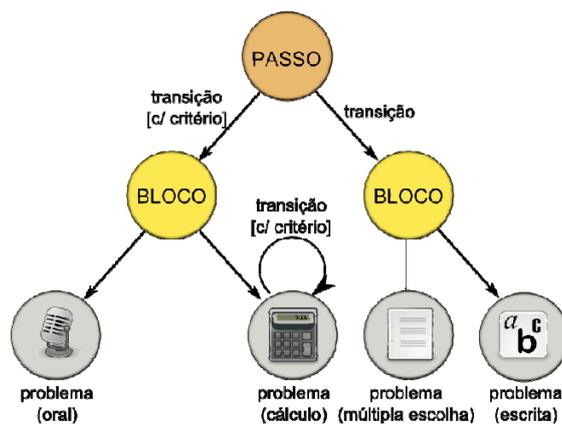
3.2. Sistemas de ensino adaptáveis

Os sistemas de ensino adaptáveis são sistemas onde o aluno ou instrutor necessita customizar e configurar a sessão de ensino de maneira manual para o estudante (Marković & Jovanović, 2011). Estes sistemas exigem que um instrutor esteja presente ao lado do aluno de maneira integral ou semi-integral para sua melhor eficácia. O instrutor 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) desenvolvido por Orlando (2009), foi utilizado em algumas cidades do interior do estado de São Paulo, como Araraquara, Ribeirão Bonito e São Carlos. Nessas cidades, o aplicativo foi empregado em escolas, a alunos com ou sem problemas da aquisição aprendizado na leitura e escrita. Pode-se considerar que o GEIC informatizou o processo do programa de ensino ALEPP (D. G. de Souza et al., 2009). A capacidade de customização e modularização de blocos de tarefas de ensino através do GEIC é um destaque do programa, como exemplifica a Figura 3.12. O sistema permite que o fluxo de ensino seja alterado conforme regras pré-estabelecidas pelo instrutor, contudo, no decorrer das sessões de ensino, verificou-se que algumas crianças perdiam a motivação

em realizar as tarefas de ensino, uma vez que estas são realizadas de forma sequencial, ininterrupta e proporcionam desinteresse (Siqueira, Barros, Monteiro, Souza, & Marques, 2012).

Figura 3.12 – Estrutura de uma Sessão de Ensino no GEIC.



Fonte: Orlando *et al.*, 2009.

É importante destacar estudos posteriores, inspirados no aplicativo GEIC, como o de Siqueira *et al.* (2012), que trataram os jogos digitais como uma interface entre as tarefas de ensino e o aluno. O objetivo foi tornar as tarefas mais interessantes e o processo de ensino mais lúdico. A Figura 3.13 mostra o jogo em aplicação.

Figura 3.13 – Uma cena do jogo desenvolvido para auxiliar no aprendizado da leitura.



Fonte: Adaptado de (Siqueira, 2012).

O artigo de Sampayo *et. al* (2013) aborda a motivação e o ajuste da dificuldade de aprendizado dos desafios apresentados através de jogo computacional e através de um texto impresso. Os resultados do pré-testes e pós-testes sugerem que os estudantes que passaram pelo jogo com ajuste de dificuldade tiveram melhores taxas de aprendizado com cerca de 13% de melhoria no aprendizado, mas com altas taxas de atraso e compreensão dos desafios mais complexos. O autor explica que um ajuste automático para a dificuldade

experimentada em jogos educacionais pode promover o aprendizado de leitura e escrita de estudantes em diferenciar cognatos do idioma espanhol, logo alguns algoritmos para esse fim poderiam ser empregados para o processo de ensino, visto que no trabalho o ajuste adaptativo de dificuldade foi feito de modo manual.

O desenvolvimento do trabalho de Ronimus *et al.* (2014) teve como objetivo auxiliar crianças com problemas de dislexia a identificarem e assimilarem os sons das letras e palavras. O estudo desenvolvido na Finlândia, e estendido para países africanos, buscou analisar os efeitos da recompensa, desafios e engajamento das crianças em jogos educacionais. O conjunto de jogos, como mostra a Figura 3.14, permite auxiliar a identificar letras básicas e a relação com o som que cada letra, progredindo gradualmente de nível entre sílabas e palavras. Da mesma forma, o estudo expõe que jogos digitais podem ser bons instrumentos para a alfabetização precoce.

Figura 3.14 – Jogo para treino de fonemas. O jogador clica na sílaba correspondente e uma animação faz o canhão atirar nos balões clicados.



Fonte: Adaptado de Richardson & Lyytinen (2014).

Os estudos como de Chin *et al.* (2013), Seiter *et al.* (2013) são alguns que utilizaram sistemas para o ensino de metodologia científica e pensamento computacional para crianças do ensino fundamental através de tarefas customizáveis. Seus experimentos foram aplicados com alunos reais, onde obtiveram constatação de potencial eficácia em detectar o raciocínio hierárquico e o pensamento computacional.

O estudo de Ekin, Cagiltay, & Karasu (2018) destacou o aspecto lúdico do ensino em especial a crianças com deficiência mental. De acordo com os resultados, brinquedos inteligentes com ações interativas e feedbacks animados têm um efeito positivo no ensino, o que é significativo no desenvolvimento de habilidades acadêmicas e sociais para indivíduos com deficiência mental moderada.

A pesquisa de J. Wang, Jou, Lv, & Huang (2018) utilizou questionários e entrevistas para investigar o ensino em sala de aula baseado em tecnologias de ensino

investigando os efeitos sobre a aprendizagem de modelos da física. As modernas tecnologias de ensino (multimídia digital e plataforma online) guiaram os alunos através das etapas da construção do novo conhecimento durante o processo de aprendizagem, fornecendo uma referência empírica para a aplicação de tecnologias de ensino.

O estudo de H.-Y. Hong & Chai (2017) investigou o desenvolvimento de práticas e crenças de ensino de matemática adaptativa em um ambiente de construção de conhecimento on-line. Os alunos demonstram progressivamente práticas de ensino mais criativas e adaptativas desenvolvendo crenças matemáticas mais orientadas para o construtivismo.

Os sistemas apresentados neste subcapítulo usam a intervenção manual para a construção de conjunto de tarefas, o que pode ser considerado uma limitação, pois o tutor necessitaria construir um determinado conjunto de tarefas para cada aluno, observando o seu desempenho e suas habilidades. O especialista (i.e. professor, instrutor ou psicólogo) é indispensável para o ensino (Nogueira & Both, 2012; Pedri, 2012), o que naturalmente torna o processo de modelagem dos blocos de ensino mais dispendioso em tempo e esforço, adicionada ainda à subjetividade na criação das tarefas ou classificação de aprendizado.

3.3. Sistemas de tutoria e aprendizado adaptativos

Os sistemas de tutoria inteligentes (*Intelligent Tutoring Systems - ITS*) e ambientes de aprendizado inteligentes (*Intelligent Learning Environments*) podem ser enquadrados em um grupo chamado de sistemas de tutoria e aprendizado adaptativos. Estes sistemas são focados na adaptação do aprendizado ao longo do tempo baseado em contínua observação de preferências e conhecimento do usuário (Marković & Jovanović, 2011). Ou seja, o perfil do usuário não é considerado estático, mas atualizado depois do acompanhamento e análise do seu comportamento.

Puzenat *et al.* (2010), na aplicação de um jogo digital, avaliou aspectos cognitivos utilizando uma rede neural artificial para medir as atividades dos usuários e determinar a idade mental de crianças. Em um dos protótipos, foram usadas tarefas de seleção e construção de palavras, mas não chegavam a relacionar a dificuldade das tarefas do jogo educacional com a técnica de inteligência artificial, ou atuar no escopo do procedimento MTS.

Li *et al.* (2010) descreve um sistema adaptativo de aprendizado através de dispositivos móveis com o objetivo de enviar materiais de ensino, recomendar conteúdos e testes adaptados ao usuário do sistema. Ren *et al.* (2010) usa uma abordagem similar com relação ao nivelamento de testes para um grupo de alunos com parâmetros estimados por especialistas e alterados dinamicamente. Similarmente, em ambos os trabalhos, não é abordado a realização do ajuste de dificuldade com o nível de conhecimento do aluno no contexto da análise comportamental.

O artigo de Cabada *et al.* (2011) propõe um sistema de tutoria inteligente auxiliado por uma rede neural de *Kohonen* para identificar o estilo de aprendizado de alunos. Dessa forma, o autor cita que é possível personalizar o aprendizado individual em um ambiente colaborativo, sem auxílio de professores no processo de treinamento. O sistema foi feito para dispositivos móveis e contou com uma boa aceitação no quesito usabilidade, mas o trabalho não deixa claro como foi a avaliação da rede neural e como foi feita a recomendação de cursos para o aprendiz.

A avaliação e integração de modelos do estudante foi destaque no trabalho de Chrysafiadi e Virvou (2012). O trabalho utilizou a técnica de lógica *fuzzy* para atualizar o modelo do estudante a cada interação em um sistema que visa o ensino de uma linguagem de programação. O modelo do estudante é definido como o processo de coleta de relevantes informações com o objetivo de inferir um estado cognitivo sobre o aluno. Esse método teve bons resultados com relação à satisfação e desempenho dos participantes da pesquisa nos quesitos efetividade e adaptabilidade, contudo o método se mostrou preso ao domínio do ensino da programação para alunos da graduação.

O trabalho de Verdú *et al.* (2012) buscou uma abordagem diferente, voltada para ambientes competitivos de aprendizagem. Os autores citam que, em seu domínio específico de competição, o ajuste de dificuldade foi feito de forma empírica com auxílio do especialista (professor) como fonte primária. O trabalho usou um sistema *fuzzy* aliado a um algoritmo genético para medir a dificuldade das questões como sendo fácil, moderado e difícil. Sua validação foi feita por três especialistas e obteve boa avaliação nos quesitos de similaridades das questões geradas, contudo a identificação prévia do nível de conhecimento do aluno e seu domínio não se encaixa no modelo de ensino proposto pelo procedimento MTS.

A pesquisa de Nerino *et al.* (2012) buscou classificar o aprendizado a partir de fórmulas matemáticas, mapeando a dificuldade no ensino da leitura através de tarefas de

emparelhamento ao modelo (tarefas MTS). O grau de conhecimento de um aluno em uma determinada palavra e o seu aprendizado foi obtido através da técnica de regressão logística e regras de formação para inferência de aprendizado. 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* baseado nos dados gerados pela técnica de aprendizado de máquina. Da mesma forma, a aplicabilidade do sistema possui uma lacuna no fato da não utilização dos fatores de dificuldade inerentes da leitura, além de não abordar possíveis regras para a criação de um repertório completo de tarefas.

Méndez e González (2013) propuseram uma abordagem para obter informações sobre as tendências de aprendizado através de um modelo que envolve o índice de participação do aluno e seu desempenho. A proposta computacional usada estimou o nível do estudante em um curso de engenharia. Um dos resultados mostrados revelou que o desempenho do aluno melhorou durante as primeiras quatro semanas do curso, mas através dos gráficos foi identificado que na quarta semana houve um declínio da performance do aluno. Por ser muito específico para o ensino de adultos e para o ensino de disciplinas das engenharias, a adaptação deste método para o procedimento MTS não se enquadraria.

Outro estudo interessante é o de Lin *et al.* (2013), que resultou em um caminho de aprendizagem personalizado baseado em árvore de decisão. Os autores sugerem que esta técnica pode ajudar e encorajar o interesse em sistemas de aprendizado através de seleção de componentes de criatividade. O resultado mostrou que uma árvore de decisão, definida através do algoritmo C4.5, foi capaz de selecionar componentes de criatividade melhorando as taxas de acerto dos estudantes universitários. No entanto, a dificuldade das tarefas de ensino para as crianças em idade escolar não foi verificada, visto que o sistema é focado para pessoas adultas.

A personalização do conteúdo de ensino também foi estudada por Chang *et al.* (2013) e Li *et al.* (2012), que utilizaram meta-heurísticas bio-inspiradas, mais precisamente, algoritmos genéticos e algoritmos de otimização por enxame de partículas. O uso de GA pode ser mais preciso para encontrar materiais para cursos em domínios mais específicos, segundo Chang *et al.* (2013), mas segundo Li *et al.* (2012) o PSO se mostra melhor quando se trata de problemas com um grande número de alunos e diferentes cursos on-line. Os autores utilizaram os algoritmos para oferecer cursos e

materiais de ensino presentes em um banco de dados de tarefas, sem o foco para o ensino da leitura via procedimento MTS.

O trabalho de Krynicky, Jaen e Navarro (2016) apresentou o conceito de reaprendizado, que tenta ajudar as pessoas a recuperar habilidades motoras e cognitivas básicas perdidas devido a doença ou acidente. Foi levado em consideração a avaliação adequada das necessidades de uma pessoa afetada, levando a um diagnóstico, seguido de uma recomendação sobre os exercícios, testes e tarefas a serem executadas. Nesta pesquisa foi utilizado um algoritmo de otimização por colônias de formigas, visto que, segundo o autor, o comportamento das formigas tem semelhanças com o processo de aprendizagem humana.

Novos trabalhos indicam que a compreensão do aprendizado em sistemas de *e-learning* podem auxiliar tutores (Holmes, Latham, Crockett, & OShea, 2018; Viswanathan & VanLehn, 2018). Estes estudos utilizam análises descritivas do comportamento do aprendiz, técnicas de processamento de imagem e redes neurais artificiais para modelar e classificar o comportamento verbal e não-verbal. Problemas matemáticos complexos podem ser tópicos de assuntos que são ensinados enquanto os dados são coletados em registros de ação do usuário em tablets. Estes trabalhos utilizam tarefas criadas manualmente, mas estão nesta categoria de “sistemas de tutoria e aprendizado adaptativos” pois utilizam técnicas que identificam características cognitivas de aprendizado, sem a intervenção direta do especialista, o que pode auxiliar na geração de tarefas de ensino.

É importante destacar que poucos dos sistemas inteligentes propostos na literatura 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 ressaltaram o uso de algum modelo de inteligência computacional como potencial proposta de otimizar algum fator da aprendizagem.

3.4. Sistemas de testes adaptativos

Alguns dos trabalhos pesquisados durante essa tese focaram em testes adaptativos computadorizados (*Computerized Adaptive Testing* - CAT), pois apesar de serem sistemas avaliativos, geralmente adicionam a noção de dificuldade em seu escopo. Sistemas CAT são usados para adaptar um teste, exame ou prova, de acordo com a

habilidade do aluno (Conejo, Guzmán, Cruz, & Barros, 2014). As pesquisas aqui analisadas são focadas em uma questão: “Como selecionar/gerar tarefas de maneira automática em testes computadorizados ou sistemas de aprendizagem?”.

No campo de ensino, aprendizagem e avaliação da aprendizagem auxiliado por computador, mais precisamente por técnicas de inteligência computacional, podemos iniciar com os trabalhos de Chen (2008) e Chen *et al.* (2008). Os autores expõem a criação de 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. Um outro módulo, baseado em lógica *fuzzy*, realiza a estimação da habilidade do aluno. 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 autor sugere que a customização é importante para promover o desempenho em testes individualizados, contudo deixa de lado outros possíveis fatores de dificuldade como a similaridade e os tipos de tarefas.

Cheng *et al.* (2008) usa regressão linear para prever a dificuldade de questões e adaptar a dificuldade das questões em CAT para sistemas multimídias. Já Baylari *et al.* (2009) faz uma revisão de sistemas de *e-learning* e ambientes de aprendizado inteligentes para a personalização. A proposta dessa pesquisa foi a utilização de um *framework* aliado à uma rede neural artificial para avaliar alunos e diagnosticar problemas de aprendizado.

O trabalho de Liu e Chen (2012) tratou de expor métodos de aprendizagem tradicionais e baseados em computador. Os autores frisam que os tutores devem desenvolver métodos de avaliação eficazes para ajudar as pessoas a entender os ganhos de estudo alcançados. Para isso, foi proposto um novo método para compor documentos de testes a partir de uma base de dados. Uma das estratégias usadas para a seleção de testes, distingue-se entre os usuários que compreendem bem os conteúdos de aprendizagem e aqueles que não possuem domínio claro sobre o material do curso. Neste intuito, foram utilizadas técnicas de mineração de dados para analisar a associação dos testes. O resultado se mostrou eficaz quando comparado com métodos tradicionais e requer menos iterações que as técnicas seleção. Contudo, existe a necessidade de um banco de tarefas previamente calibrados. A proposta requer grande variação dos tipos de testes a fim de cobrir um maior campo de conteúdos e tópicos de ensino em uma

abordagem, o que é feito de modo subjetivo e custoso, pois diversos especialistas necessitariam adicionar tarefas no banco.

O trabalho de Özyurt *et al.* (2012) apresenta um módulo de teste adaptativo para suporte de estimação de habilidade e testes de tópicos relacionados com a matemática. Eles utilizam a adaptação automática de testes conforme o estilo de aprendizado do aluno. Os autores sugerem que a avaliação individual melhorou com as técnicas utilizadas, em comparação com os testes clássicos. Um banco de questões é utilizado, mas não fica claro como as questões foram modeladas, dando a entender que os pesquisadores criaram manualmente as questões para os tópicos propostos.

Shen *et al.* (2012) apresentaram uma ferramenta para auxiliar testes adaptados por computador na avaliação de sistemas de pontuação em provas. Seu método foi a utilização da lógica *fuzzy* para avaliar testes quando tarefas são incertas, por exemplo: completamente correto, parcialmente correto, parcialmente incorreto e completamente incorreto. Seus resultados mostraram que o sistema pode inferir o índice de desempenho das respostas e prever qual a nota final do aluno, o que não ocorre em sistemas de avaliação tradicional. Uma das desvantagens é que seu escopo está limitado a sistemas de avaliação de provas, testes e exames, onde os resultados das questões são incertos; não sendo aplicável à maioria dos testes comportamentais, que utilizam repostas 100% corretas ou incorretas.

Já Millán *et al.* (2013) usou Redes Bayesianas como modelo do estudante em um ambiente de teste computadorizado para avaliar o conhecimento de matemática dos alunos. O modelo identifica se o aluno possui conhecimento em determinadas tarefas. Os parâmetros de entrada foram definidos por especialistas, onde as questões possuem duas respostas: verdadeiro ou falso. Em média, o sistema mostrou confiabilidade em estimar o conhecimento do aluno através de experimentos realizados por computador e testes impressos. O sistema possui algumas limitações, como o uso específico para a avaliação da aritmética, e ainda a falta de interatividade e receptividade do sistema por parte dos alunos.

Conejo *et al.* (2014) conduziu um estudo empírico que explora a relação entre a mensuração de dificuldades de questões de ensino por professores, alunos e valores estimados de dados experimentais através do TRI e TTC (Teoria de Testes Clássicos). Os sistemas que foram examinados buscaram emular o comportamento de um professor como um processo de tutoria de aprendizado, conceitualmente chamados de *Adaptive and*

Intelligent Web-Based Educational Systems. Conforme Conejo *et al.* (2014), esses sistemas são uma evolução de *Intelligent Tutoring Systemns* (ITS) e *Adaptive Hypermedia Systemns*, mas voltados para a internet. Os autores indicam que alunos são melhores estimadores de dificuldade que professores; ou seja, nem sempre agentes humanos são melhores indicadores para dificuldade de uma tarefa de ensino. O autor cita que utilizou tarefas com baixa complexidade e que a utilização de tarefas com maiores dificuldades poderia interferir nos experimentos.

Em relação aos sistemas de testes adaptativos apresentados neste subcapítulo, observou-se a não utilização de dois ou mais fatores de dificuldades. Em uma tarefa é possível que existam não somente um valor de dificuldade, mas diversos níveis que influenciem na decisão correta do aluno. No domínio de aplicação desta tese, este problema específico é tratado, considerando dois fatores de dificuldade em uma só tarefa, onde uma tarefa possui: dificuldade de similaridade entre escolhas; e dificuldade dos tipos de tarefas, além de levar em consideração o acerto da tarefa ao acaso.

3.5. Análise dos trabalhos pesquisados

Esta tese avaliou as heurísticas criadas pelos autores apresentados visando à adaptação ou criação de novas heurísticas para o ensino da leitura, utilizando procedimentos de equivalência de estímulos para o público infantil. Pretende-se preencher as lacunas apontadas nos estudos, na utilização de meios e dispositivos de aprendizagem personalizada. Um sumário é exibido na Tabela 3.4 buscando facilitar a interpretação dos estudos relacionados.

Tabela 3.4 – Sumário de trabalhos analisados. Sistemas de ensino adaptáveis: sistemas educacionais auxiliados por um especialista. Sistemas de tutoria e aprendizado adaptativos: sistemas educacionais auxiliados por técnicas de inteligência computacional. Sistemas de testes adaptativos: sistemas educacionais focados na avaliação do aprendizado através de técnicas de inteligência computacional.

Trabalho	Técnica de IC utilizada	Sistema educacional	Principal Contribuição	Limitação
(Souza D. G., et al., 2009)	---	Sistemas de ensino adaptáveis	Gerou tarefas para verificar o ensino-aprendizado da leitura a crianças. O programa utilizado foi o ALEPP.	Geração manual de tarefas de ensino.
(Orlando, 2009)	---	Sistemas de ensino adaptáveis	Criou um sistema de gerenciamento de tarefas de ensino.	Geração manual de tarefas de ensino.
(Siqueira, 2012)	---	Sistemas de ensino adaptáveis	Buscou incluir aspectos de ludicidade as tarefas de ensino do ALEPP	Geração manual de tarefas de ensino.
(Sampayo, Cope, He & Byrne, 2013)	---	Sistemas de ensino adaptáveis	Ajustou a dificuldade das tarefas manualmente buscando encontrar a correlação entre motivação e aprendizado	Geração manual de tarefas de ensino.
(Chin, Dohmen, & Schwartz, 2013)	---	Sistemas de ensino adaptáveis	Criado para ensinar o raciocínio hierárquico e utilizado para o ensino da ciência	Geração manual de tarefas de ensino.
(Seiter & Foreman, 2013)	---	Sistemas de ensino adaptáveis	Sistema aplicado para alunos de primeira a sexta série para o desenvolvimento do pensamento computacional	Geração manual de tarefas de ensino.
(Ronimus, Kujala, Tolvanen, & Lyytinen, 2014)	---	Sistemas de ensino adaptáveis	Verificou o efeito de engajamento, motivação e desafio em um jogo educacional para crianças	Geração manual de tarefas de ensino.
(Eliasquevici, et al., 2014)	---	Sistemas de ensino adaptáveis	Buscou incluir aspectos de ludicidade as tarefas de ensino do ALEPP com maior interatividade e imersão.	Geração manual de tarefas de ensino.
(Ekin, Cagiltay, & Karasu 2018)	---	Sistemas de ensino adaptáveis	Buscou ensinar instruções através de brinquedos interativos.	Geração manual de tarefas de ensino.

Trabalho	Técnica de IC utilizada	Sistema educacional	Principal Contribuição	Limitação
(J. Wang, Jou, Lv, & Huang, 2018)	---	Sistemas de ensino adaptáveis	Utilizou técnicas de ensino modernas (multimídia digital e plataforma online) para ensinar física.	Geração manual de tarefas de ensino.
(H.-Y. Hong & Chai, 2017)	---	Sistemas de ensino adaptáveis	Investigou-se o desenvolvimento de práticas e crenças de ensino de matemática por parte de alunos em um sistema on-line	Geração manual de tarefas de ensino.
(Puzenat & Verlut, 2010)	Rede Neural Artificial	Sistemas de tutoria e aprendizado adaptativos	Identificou o comportamento e inferiu a idade mental de crianças.	Restrito ao seu domínio de aplicação, ou seja, focou em um domínio específico de ensino, sem possibilidade para adequação ao problema de geração de tarefas.
(Li, et al., 2010)	---	Sistemas de tutoria e aprendizado adaptativos	Recomenda conteúdos e testes adaptados através de dispositivos moveis	Restrito ao seu domínio de aplicação. Sem abordagem sobre o ajuste de dificuldade
(Ren, Liu, Ye, Wang, & Wang, 2010)	---	Sistemas de tutoria e aprendizado adaptativos	Altera os parâmetros oferecidos por especialistas para o nivelamento de testes a um grupo de alunos em um sistema de ensino.	Restrito ao seu domínio de aplicação. Não leva em consideração o ensino individualizado e sim nivela o ensino a um grupo de alunos.
(Cabada, Barrón, & Reyes, 2011)	Rede neural artificial de Kohonen	Sistemas de tutoria e aprendizado adaptativos	Identificou agrupamentos de alunos conforme o seu estilo de aprendizado.	Lacuna no que se refere à avaliação da técnica de IC.
(Chrysafiadi & Virvou, 2012)	Lógica Fuzzy	Sistemas de tutoria e aprendizado adaptativos	Utilização e avaliação de um modelo e estudante através da técnica de IC.	Restrito ao seu domínio de aplicação: para alunos do ensino superior e ensino de programação.

Trabalho	Técnica de IC utilizada	Sistema educacional	Principal Contribuição	Limitação
(Verdú, Verdú, Regueras, De Castro, & García, 2012)	Algoritmo genético e lógica fuzzy	Sistemas de tutoria e aprendizado adaptativos	Identificação de dificuldade de questões para sistemas competitivos de aprendizado.	Restrito ao seu domínio de aplicação. Não utilizou outros fatores de dificuldade.
(Souza, Pereira, Cavalcante, Sales, & Marques, 2012)	Regressão Logística	Sistemas de tutoria e aprendizado adaptativos	Classificação e identificação do grau de aprendizado de palavras utilizando tarefas MTS.	Experimentos com dados simulados. Não utilizou outros fatores de dificuldade.
(Pereira et al., 2013)	Lógica Fuzzy	Sistemas de tutoria e aprendizado adaptativos	Sugestão de tarefas de ensino do tipo MTS.	Experimentos com dados simulados. Não utilizou outros fatores de dificuldade. Lacuna no que se refere a geração de um repertório completo de tarefas.
(Méndez & González, 2013)	Lógica Fuzzy	Sistemas de tutoria e aprendizado adaptativos	Predição de tendências de desempenho de aprendizado e ensino em cursos de engenharia.	Restrito ao domínio de aplicação.
(Chun, Yehb, Hsin, & I, 2013)	Árvore de Decisão: C4.5	Sistemas de tutoria e aprendizado adaptativos	Identificação de caminhos de aprendizagem baseado em componentes de criatividade.	Não mensurou os fatores de dificuldade das atividades de ensino. Restrito ao domínio de aplicação.
(J.-W. Li et al., 2012)	Algoritmo de otimização por enxame de partículas e algoritmos genéticos	Sistemas de tutoria e aprendizado adaptativos	Personalização de cursos on-line para grupos de alunos.	Restrito ao domínio de aplicação.

Trabalho	Técnica de IC utilizada	Sistema educacional	Principal Contribuição	Limitação
(T. Y. Chang & Ke, 2013)	Algoritmo de otimização por enxame de partículas e algoritmos genéticos	Sistemas de tutoria e aprendizado adaptativos	Composição de cursos e materiais personalizados na web.	Não mensurou os fatores de dificuldade das atividades de ensino. Restrito ao domínio de aplicação.
(Krynicky et al., 2016)	Otimização por colônia de formiga	Sistemas de tutoria e aprendizado adaptativos	Utilizou o conceito de reaprendizado para pessoas que sofreram lesões cerebrais e testou seu método para vários cenários de aprendizado.	Não mensurou os fatores de dificuldade das atividades de ensino. Restrito ao domínio de aplicação.
(Holmes et al., 2018; Viswanathan & VanLehn, 2018)	Rede Neurais	Sistemas de tutoria e aprendizado adaptativos	Utilizam análises descritivas do comportamento verbal e não-verbal de aprendizes para auxiliar no ensino de problemas matemáticos.	Não realiza a geração automática de tarefas de ensino mas trabalham com o comportamento verbal de alunos. Restrito ao domínio de aplicação.
(C.-M. Chen, 2008)	Algoritmo Genético	Sistemas de testes adaptativos	Recomendação de caminho de aprendizado para testes em cursos baseado no TRI.	Restrito ao Domínio de aplicação: para recomendação de testes de cursos. Não mensurou os fatores de dificuldade das atividades de ensino.
(C.-M. Chen & Duh, 2008)	Lógica Fuzzy	Sistemas de testes adaptativos	Identificação da habilidade do aluno com auxílio do modelo do estudante e baseado no TRI para testes em cursos.	Restrito ao Domínio de aplicação: identificação da habilidade do aluno. Não mensurou os fatores de dificuldade das atividades de ensino.
(Cheng, Shen, & Basu, 2008)	Regressão Linear	Sistemas de testes adaptativos	Predição das dificuldades de questões em um teste computadorizado e adaptação das questões para ambientes multimídias.	Restrito ao Domínio de aplicação para testes de questões de matemática.

Trabalho	Técnica de IC utilizada	Sistema educacional	Principal Contribuição	Limitação
(Baylari & Montazer, 2009)	Rede Neural Artificial	Sistemas de testes adaptativos	Boa taxa de precisão (83%) na recomendação de testes para um pós-teste computadorizado.	Restrito ao Domínio de aplicação: criação de pós-testes.
(Wauters, Desmet, & Noortgate, 2012)	---	Sistemas de testes adaptativos	Pesquisa comparativa entre estimadores de dificuldade de testes computadorizados com o objetivo de encontrar alternativas para estimadores tradicionais.	Lacuna no que se refere a diversos fatores de dificuldade de um teste.
(Liu & Chen, 2012)	Regras de Associação: Apriori	Sistemas de testes adaptativos	Composição de um documento de testes.	Necessidade da utilização de diversos testes em uma base com calibração através de agentes humanos ou técnicas custosas computacionalmente
(Özyurt, Özyurt, Bak, & Güven, 2012)	---	Sistemas de testes adaptativos	Apresenta um módulo adaptativo de avaliação de matemática. Sugere que a avaliação individual pode ser melhor que a avaliação de testes clássicos.	Lacuna no quesito geração de tarefas.
(Shen, Yang, Wang, & Lin, 2012)	Lógica Fuzzy	Sistemas de testes adaptativos	Avaliação dos testes baseado no TRI, quando os resultados dos testes possuem valores incertos. Predição de notas de alunos.	Restrito ao Domínio de aplicação para testes com resultados incertos.
(Millán, Descalço, Castillo, Oliveira, & Diogo, 2013)	Redes Bayesianas	Sistemas de testes adaptativos	Utilização de um modelo de estudante para avaliação e estimação de habilidades do aluno com boas taxas de confiabilidade.	Restrito ao Domínio de aplicação para testes matemáticos com 2 opções. Pouca receptividade e interatividade de alunos.
(Conejo, Guzman, Perez & Barros, 2014)	---	Sistemas de testes adaptativos	Estudo comparativo entre estimadores de dificuldade de testes computadorizados para sistemas web.	Lacuna no que se refere a diversos fatores de dificuldade de um teste.
(Klerk, Veldkamp, & Eggen, 2015)	Redes Bayesianas	Sistemas de testes adaptativos	Realizou uma revisão de psicométricas em avaliações baseadas em simulações em jogos. Modelou uma rede bayesiana para realizar avaliações em ambientes de simulações	Restrito a avaliação e não ao ensino ou gerações de tarefas.

3.6. Considerações finais do capítulo

Neste capítulo foram expostos os trabalhos correlatos com a presente tese, destacando-se algumas limitações de estudos envolvendo a área de domínio. As pesquisas na literatura da área possuem objetivos de ensino para diversos fins. O presente trabalho procura contribuir cientificamente com estas características abordadas, buscando remediar os pontos fracos e estabilizar os pontos fortes apontados na Tabela 3.4.

A utilização da pesquisa desenvolvida neste trabalho como método integrado a sistemas de ensino-aprendizado almeja auxiliar, tanto aluno quanto o professor. A criação de tarefas de ensino baseado em um modelo comportamental pode ser integrada em sistemas geradores de repertórios de tarefas de ensino, a fim de auxiliar no fluxo de aprendizado, aumentando o interesse do aluno e a personalização do ensino. As lacunas identificadas nas pesquisas apresentadas serviram de base para o desenvolvimento desta tese, que se encontra com maiores detalhes no próximo capítulo.

4. Geração de tarefas de ensino adaptadas através de algoritmos bio-inspirados para crianças em fase inicial da alfabetização

4.1. Considerações iniciais do capítulo

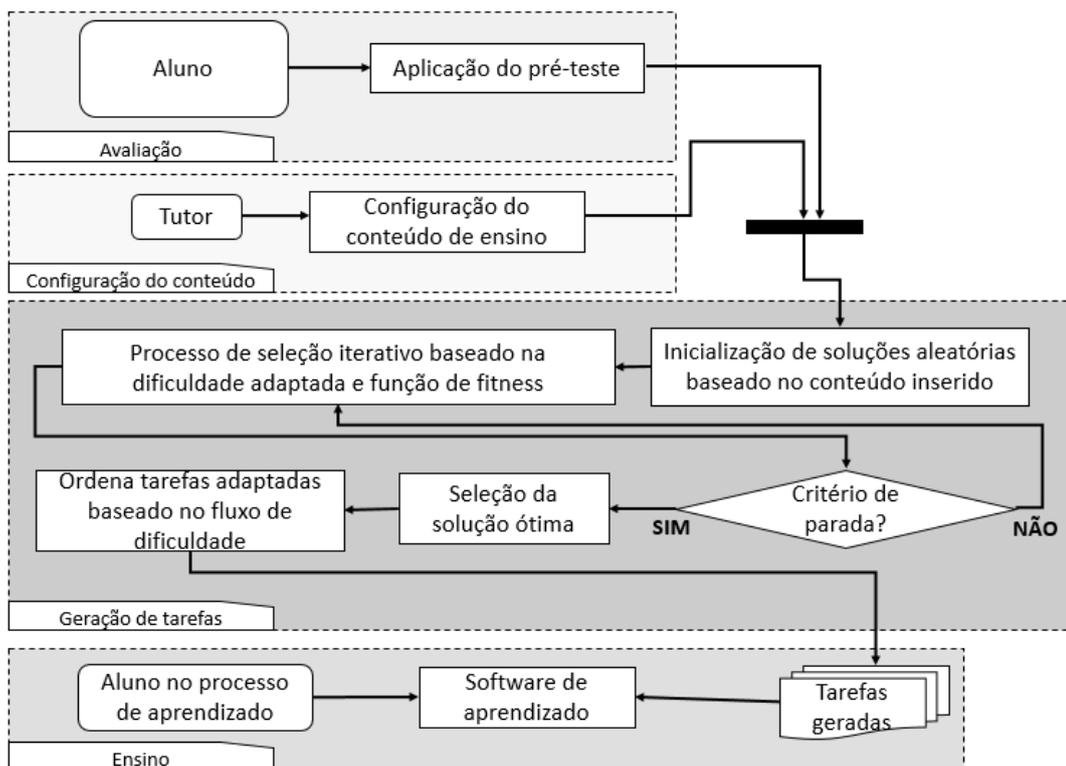
Diversos trabalhos sugerem que a criação de tarefas de ensino para alunos deve seguir um ensino individualizado (Galvão & Barros, 2001; Skinner, 1950); ademais, vários programas computacionais (Orlando, 2009; Ronimus et al., 2014) buscam auxiliar no ensino da leitura de indivíduos com alguma limitação de aprendizado. Porém, criar um conjunto de tarefas de ensino individualizado não é trivial, visto que a demanda de tempo e esforço para o educador é grande, neste caso requerendo a criação de um conjunto de tarefas para cada aluno, observando suas habilidades e limitações de aprendizado.

Logo, para a montagem de um repertório de tarefas de ensino eficaz, faz-se necessário identificar uma tarefa de ensino que melhor se adapte ao aluno (Feldman, Monteserin, & Amandi, 2014; Marković & Jovanović, 2011; Millán et al., 2013), identificando fatores de “dificuldade”, visto que este é um componente importante no aprendizado de um conteúdo (Conejo, Guzmán, Cruz, et al., 2014).

4.2. Sistema para a geração de tarefas

A presente tese apresenta o desenvolvimento de um sistema para a geração automática de um conjunto de tarefas de ensino baseado em dificuldade adaptativa e no procedimento *Matching-to-sample*. Este trabalho pode ser considerado um método adaptativo, pois gera tarefas adaptadas às necessidades e habilidades de alunos em fase inicial de aprendizado da leitura (Kostolányová & Šarmanová, 2014). Conforme apresentado no fluxograma da Figura 4.15, este sistema é dividido em quatro partes principais: Avaliação; Configuração do conteúdo; Geração de tarefas; e Ensino.

Figura 4.15 – Fluxograma geral para a geração de tarefas adaptadas



Inicialmente, o aluno é submetido a um pré-teste no processo de avaliação. Após a “Aplicação do pré-teste”, um arquivo chamado “Tarefas respondidas” é formatado e usado para calcular o conhecimento do aluno inicial para determinar o grau de dificuldade das tarefas.

No processo de configuração de conteúdo, o “Tutor” usa uma interface para “Configurar o conteúdo de ensino”, referente aos atributos das tarefas que serão geradas, que são:

- Modelos: uma lista de palavras que devem ser aprendidas;
- Tipos de tarefas: uma lista contendo todos os tipos de tarefas possíveis para a sessão de ensino;
- Escolhas: uma lista com todas as palavras que podem ser escolhidas para comparação com a lista de palavras modelos;
- Fluxo de dificuldade: um vetor que representa a “curva de dificuldade” do conjunto de tarefas que serão geradas, onde cada posição é um valor de dificuldade.

No módulo para “Geração de tarefas”, o processo adaptativo começa depois de receber os dados das etapas anteriores. Os dados inseridos (modelos, tipos de tarefas, escolhas e fluxo de dificuldade) são usados para criar aleatoriamente as soluções iniciais dos algoritmos bio-inspirados, a saber: a população inicial para o GA; o enxame inicial para o ICPSO; e o grafo

para o ACO. As meta-heurísticas GA, ICPSO e ACO geralmente funcionam de maneira semelhante: inicialização de soluções e alguns parâmetros aleatórios; execução de um processo de seleção iterativo baseado em uma função objetivo; e seleção de uma solução que melhor represente o resultado ideal.

Depois de selecionar a solução ótima, a execução das tarefas é ordenada pelas dificuldades adaptadas, de forma a corresponder às dificuldades do fluxo de dificuldade definido. Isso ocorre como um ajuste fino para corrigir algumas tarefas que podem estar em posições diferentes relativas ao fluxo de dificuldade. O Algoritmo do Apêndice I apresenta este processo.

Por fim, um “Software de aprendizado” apresenta as “Tarefas geradas” para o “Aluno no processo de aprendizado”, iniciando o processo de “Ensino”. Os dados coletados serão armazenados em uma base de dados servindo de insumo para posteriores análises.

Para a utilização das técnicas de inteligência computacional proposta neste trabalho, uma estrutura de dados para tarefas de ensino foi definida, representada no conjunto de tarefas da Tabela 1.1.

Tabela 4.5 – Exemplo de um conjunto de tarefas de ensino. ID = identificador numérico da tarefa.

ID	Tipo de Tarefa	Modelo	Escolha1	Escolha2	Escolha3	Escolha4	(...)	Escolha10	Resultado
1	AB	bolo	tubo						1
2	AC	apito	bico	pipa					0
3	BC	fita	tatu	Tomate					1
4	CB	muleta	pato	Tomate	apito				0

Assim, os atributos das tarefas utilizados neste trabalho são:

1. **Tipo de tarefa:** esse campo representa a relação estabelecida entre os estímulos. Nele identificam-se os procedimentos e os formatos dos estímulos (sons, figuras ou textos). Os tipos de tarefas são atributos nominais e estão representados na Tabela 2.2 com nomenclatura comumente utilizada por analistas comportamentais.
2. **Modelo:** o estímulo modelo é um atributo nominal que equivale à palavra que deve ser ensinada ao aluno. Em uma tarefa há no mínimo uma escolha, referente ao próprio modelo.
3. **Escolhas:** os estímulos de comparações ou estímulos de escolhas, são estímulos que funcionam como opções para testar a capacidade de identificação em relação

ao estímulo modelo. Uma tarefa pode ter diversas escolhas que são nomeadas como: Escolha1, Escolha2, Escolha3 e assim por diante.

4. **Resultado:** indica o acerto ou erro do aluno. Igual a 1, se o aluno responde corretamente, ou 0, caso a resposta seja incorreta.

Adicionalmente, considera-se para estrutura de dados que as posições das opções são posicionadas randomicamente na tela de um software de aprendizado que exibe essas tarefas, como sugere os trabalhos pesquisados por Gerard *et al.* (2014). Isto diminui a chance de que uma mesma opção correta apareça em um mesmo local da tela e aumente a possibilidade de acerto desta opção, configurando em um acerto da tarefa acidental ou ao acaso.

4.2.1. Dificuldade das tarefas de ensino

A dificuldade da tarefa de ensino apresentada neste trabalho pode ser considerada um valor adaptativo influenciado pelo desempenho do aluno. O modelo adaptativo foi proposto para determinar a dificuldade com base no conhecimento do aluno e nos fatores de dificuldade de uma tarefa. A Equação 4.9 foi inspirada e adaptada do conhecido modelo logístico Rasch (Wauters, Desmet, & Van Den Noortgate, 2012) apresentada no Capítulo 2.

$$D(\theta_{t_{\kappa}}, \theta_{s_{\eta}}, \beta_{t_{\kappa}}, \beta_{s_{\eta}}, n_c) = 1 - \left(\frac{1}{n_c} + \frac{1 - \frac{1}{n_c}}{1 + e^{-1.7((\theta_{t_{\kappa}} + \theta_{s_{\eta}}) - (\beta_{t_{\kappa}} + \beta_{s_{\eta}}))}} \right) \quad (4.9)$$

Onde:

D : é a dificuldade adaptativa da tarefa, sempre no intervalo entre 0 e 1;

κ : indexa o tipo de tarefa, que pode ser AB, AC, BC, CB, BB e CC;

η : indexa o intervalo de similaridade da tarefa, que pode ser [0, 0,2[, [0,2, 0,4 [, [0,4, 0,6 [, [0,6, 0,8 [, [0,8, 1], representando os índices I1, I2, I3, I4 e I5, respectivamente;

$\beta_{t_{\kappa}}, \beta_{s_{\eta}}$: são os fatores de dificuldade do tipo de tarefa e fator de dificuldade da similaridade da tarefa, respectivamente;

$\theta_{t_{\kappa}}, \theta_{s_{\eta}}$: são o conhecimento do aluno para o tipo de tarefa e para o intervalo de similaridade da tarefa, respectivamente;

n_c : é o número de opções da tarefa;

O conhecimento do aluno $(\theta_{t_{\kappa}}, \theta_{s_{\eta}})$ e os fatores de dificuldade $(\beta_{t_{\kappa}}, \beta_{s_{\eta}})$ são armazenados em um banco de dados. A dificuldade do tipo de tarefa $\beta_{t_{\kappa}}$, é definida por um especialista em uma configuração prévia. Os parâmetros $(\theta_{t_{\kappa}}, \theta_{s_{\eta}}, \beta_{t_{\kappa}}, \beta_{s_{\eta}})$ são normalizados de acordo com o *Min-Max* (Fazal Rehman Shamil, 2018) para o uso correto da Equação 4.9.

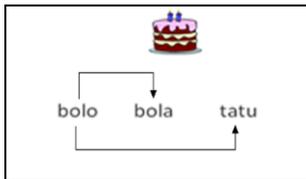
A similaridade entre as escolhas pode ser considerada um fator na avaliação da dificuldade da tarefa (Hanna et al., 2010; Nerino G Jr, Fontes M, Monteiro D, & Santana A, 2016). Neste estudo, o valor de similaridade entre as escolhas é calculado fazendo-se uma comparação entre o modelo de referência e as escolhas da tarefa. A definição do fator de dificuldade de similaridade da tarefa é composta pela similaridade entre as escolhas, representada na Equação 4.10, onde, s_i é a similaridade entre as escolhas:

$$\beta S = \frac{\sum_{i=1}^{n_c-1} s_i}{n_c-1} \quad (4.10)$$

A tarefa da Figura 4.16 exemplifica este processo, onde o modelo é “bolo” em formato de uma imagem e as escolhas são “bola” e “tatu” em formato textual. Na tarefa, é necessário comparar essas escolhas com o modelo “bolo”, como mostrado pelas setas na Figura 4.16 (a). No exemplo (Figura 4.16 (b)), a semelhança da palavra “bolo” com a palavra “bola” é 0,8, e com a palavra “tatu” é 0,1, resultando em um $\beta S = 0,45$.

Figura 4.16 – Exemplo do método para calcular a similaridade entre escolhas de uma tarefa. (a) exemplo de uma tarefa BC, onde as setas representam a “ação de comparação” e o cálculo da similaridade; (b) o cálculo do fator de similaridade para a tarefa em questão.

Quantidade de similaridades ($n_c - 1$)	Primeira escolha	Segunda escolha	Similaridade (S_i)	Fator de similaridade da tarefa (βS)
2	"bolo"	"bola"	0.8	$\beta S = \frac{(0.8 + 0.1)}{2}$
	"bolo"	"tatu"	0.1	



(b)

A similaridade (s_i) entre escolhas textuais é calculada através de um algoritmo de similaridade que atribui pesos a algumas regras; por exemplo, às sílabas iniciais ou finais de uma palavra (Nerino et al. 2016). Isso ocorre porque, geralmente, a criança faz a seleção observando a primeira ou a última sílaba da palavra (Muller et. al. 2000). Esses pesos foram definidos após algumas consultas com especialistas que trabalham com este processo de ensino, permitindo a identificação de um percentual de similaridade entre as palavras. Neste trabalho, o algoritmo de similaridade textual está apresentado no Apêndice II.

A técnica de Wang *et al.* (2004) foi proposta para calcular a similaridade (s_i) de pixels em imagens, usado para estimar a similaridade de tarefas que envolvem escolhas visuais. Para melhorar o desempenho, essa abordagem exige que as imagens tenham dimensões iguais e que os objetos estejam centralizados.

Os parâmetros θt_{κ} e θs_{η} são calculados com base no número de tarefas realizadas no pré-teste, e representam o conhecimento do aluno para cada um dos fatores de dificuldade de uma tarefa (tipo de tarefa e similaridade da tarefa). O θs_{η} é definido calculando todos os valores de similaridade das tarefas (βs_{η}) que foram respondidas. Os valores de similaridades são então organizados e discretizados uniformemente em cinco intervalos; assim, a porcentagem de tarefas respondidas corretamente para cada intervalo de similaridade (I1, I2, I3, I4 e I5) é determinada e usada para definir θs_{η} . O parâmetro θt_{κ} é baseado na porcentagem de tipos de tarefas que foram respondidas corretamente. As Equações 4.11 e 4.12 mostram os cálculos para esses parâmetros.

$$\theta t_{\kappa} = r_{min} + \left(\frac{Ct_{\kappa}}{Qt_{\kappa}} \right) \times (r_{max} - r_{min}) \quad (4.11)$$

$$\theta s_{\eta} = r_{min} + \left(\frac{Cs_{\eta}}{Qs_{\eta}} \right) \times (r_{max} - r_{min}) \quad (4.12)$$

Onde,

Qt_{κ} : é o número de tarefas respondidas do tipo κ ;

Ct_{κ} : é o número de tarefas corretamente respondidas do tipo κ ;

Qs_{η} : é o número de tarefas respondidas com intervalos de similaridade η ;

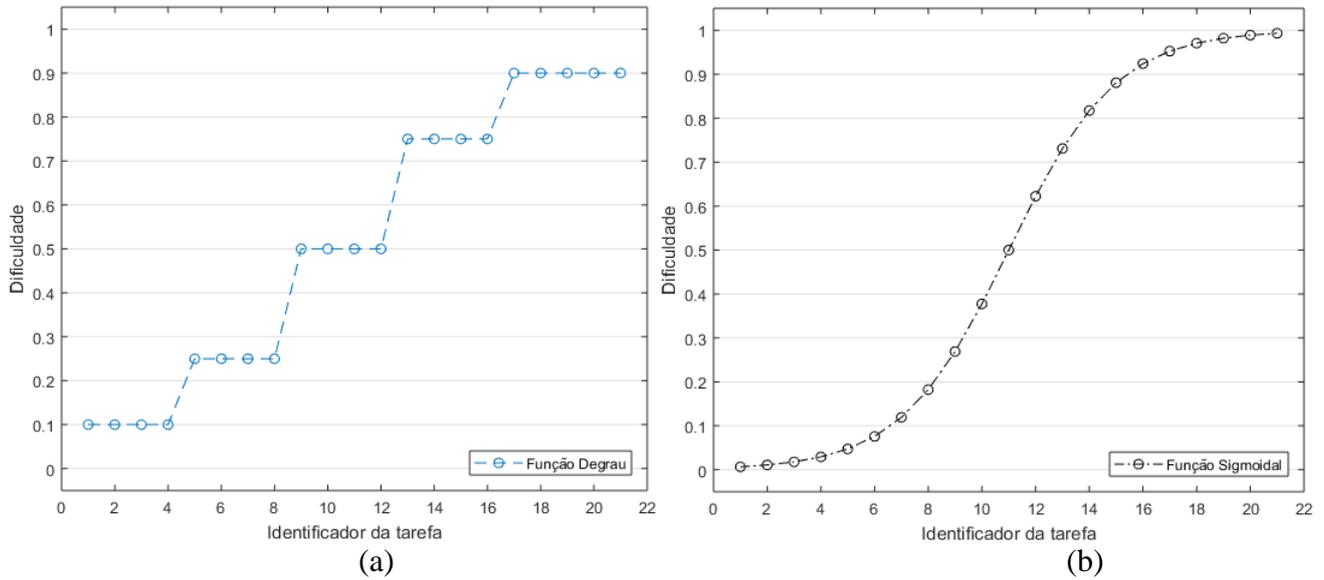
Cs_{η} : é o número de tarefas respondidas corretamente com intervalos de similaridade η ;

r_{min} , r_{max} : é o intervalo máximo e mínimo usados para normalizar θt_{κ} e θs_{η} , onde r_{min} e r_{max} são -1,5 e +1,5 adquiridos após diversos testes sobre a Equação 4.9 viabilizando a produção de valores de dificuldades para as tarefas de ensino.

4.2.2. Função de fitness

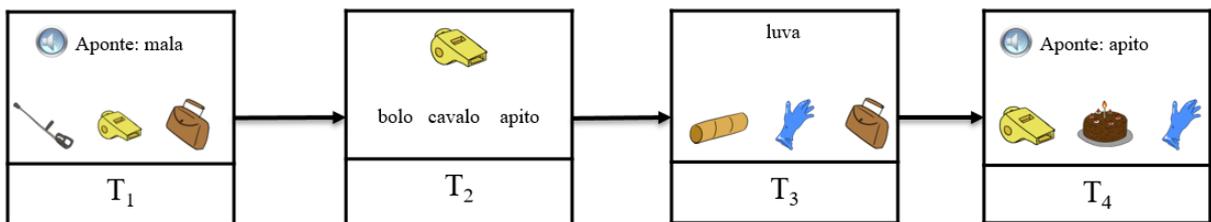
O problema na geração de um conjunto de tarefas se baseia na necessidade de determinar os valores de dificuldade de cada tarefa e na adaptação desses valores para cada aluno. Ao realizar isso automaticamente, o instrutor pode configurar um conjunto de valores de dificuldade e buscar tarefas que estejam próximas a um conjunto de dificuldades determinados anteriormente. Por exemplo, o instrutor pode usar um fluxo de dificuldade para gerar tarefas mais fáceis no início da atividade de ensino e depois passar para tarefas mais difíceis com um comportamento semelhante aos “degraus de uma escada” (Figura 4.17(a)), ou pode recomendar uma função sigmoideal, com tarefas mais fáceis, mas que crescem suavemente em dificuldade (Figura 4.17(b)).

Figura 4.17 – Exemplos de fluxo de dificuldade que podem ser definidos no sistema.



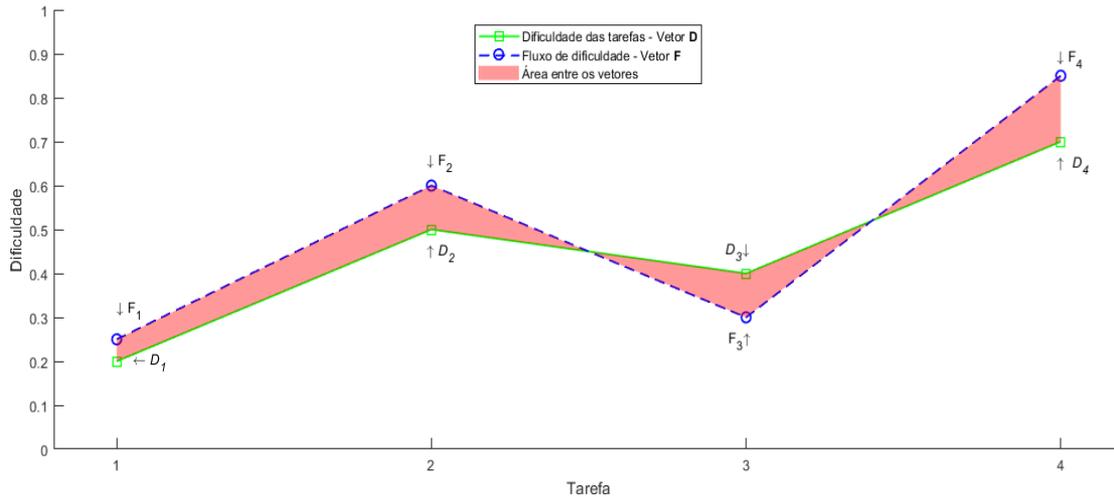
Considere o exemplo do conjunto de tarefas mostrado na Figura 4.18, em que T_1 é a primeira tarefa que será apresentada ao aluno. Nesta tarefa, o modelo é “mala”, palavra falada (formato sonoro), e as opções estão em formato de imagens. T_2 é a segunda tarefa com o modelo em formato de imagem e as opções são palavras (bolo, cavalo, apito) em formato textual; T_3 e T_4 seguem a mesma lógica.

Figura 4.18 – Exemplo de um conjunto de tarefas.



Cada tarefa tem uma dificuldade adaptada que forma um vetor $\mathbf{D} = [D_1, D_2, D_3, \dots, D_m]$, onde \mathbf{D} é a dificuldade e m é o identificador do último elemento. O fluxo de dificuldade, configurado pelo instrutor, tem uma estrutura semelhante e é representado por um vetor $\mathbf{F} = [F_1, F_2, F_3, \dots, F_m]$. A distância entre os vetores pode ser observada na Figura 4.19, onde a distância entre o fluxo de dificuldade e as dificuldades de tarefas adaptadas pode ser determinada pela área entre os dois vetores.

Figura 4.19 - Fluxo de dificuldade e a dificuldade das tarefas.



Quanto menor a área, menor a distância entre os vetores, ou seja, o fluxo de dificuldade (**F**) está mais próximo da dificuldade do conjunto de tarefas (**D**). Assim, o objetivo é encontrar o conjunto ideal de tarefas que reduzem essa área. A Equação 4.13 constitui o cálculo necessário para atribuir um valor de *fitness* a um conjunto de tarefas.

$$f(\mathbf{D}, \mathbf{F}) = 4N_T N_{max} - (A(\mathbf{D}, \mathbf{F}) + \sum_{j=1}^4 \Phi_j) \quad (4.13)$$

Onde,

f : é o valor de fitness;

D: é o vetor com a dificuldade adaptada para um conjunto de tarefas;

F: é o fluxo de dificuldade, ou seja, o vetor de dificuldade que se deseja apresentar para o aluno e que é incluído no sistema previamente;

$A(\mathbf{D}, \mathbf{F})$: é a área entre o vetor **D** e **F**;

N_T : é o número total de tarefas que deseja-se gerar para uma sessão de ensino;

N_{max} : é o número total de escolhas para geração conjunto de tarefas;

Φ_j : refere-se a penalidade de índice j , onde o número 4 refere-se ao número de penalidades definidas na Tabela 4.6.

Em um conjunto de tarefas, há a necessidade de diversidade de atributos das tarefas. Neste trabalho, busca-se evitar a repetição de tarefas com os mesmos atributos (ou seja, os estímulos que compõem as escolhas e o tipos de tarefas). No processo iterativo, para diminuir a chance de repetir atributos em tarefas do mesmo conjunto, algumas restrições e penalidades dinâmicas (A. E. Smith & Coit, 1996) são aplicadas à função de *fitness*, determinadas na Tabela 4.6.

Tabela 4.6 – Descrição das restrições e suas penalidades atribuídas ao valor de *fitness*.

Penalidades	Descrição
$\Phi_1 = \frac{n_R * A(D, F)}{N_T}$	Penalidade pelas tarefas idênticas e repetidas no mesmo conjunto de tarefas. Está é a primeira restrição a ser avaliada, onde as tarefas identificadas como repetidas não entrarão no cálculo das penalidades Φ_2 , Φ_3 e Φ_4 . n_R , é o número de tarefas idênticas no conjunto.
$\Phi_2 = \frac{n_{TM} * A(D, F)}{2 * N_T}$	Penalidade por repetição de tipos de tarefas e modelos idênticos em um conjunto de tarefas. Esta é a segunda restrição a ser avaliada, onde as tarefas que foram identificadas como repetidas não entrarão no cálculo de penalidades de Φ_3 e Φ_4 . O denominador é multiplicado por 2 referindo-se aos atributos modelo e tipo de tarefa; n_{TM} , é o número de tarefas com tipo e modelos idênticos em um conjunto de tarefas.
$\Phi_3 = \frac{n_M * A(D, F)}{n_a * N_T}$	Penalidade por repetição de modelos idênticos, onde: n_M , é o número de tarefas com modelos idênticos em um conjunto de tarefas; n_a , é o número de atributos da tarefa.
$\Phi_4 = \frac{n_T * A(D, F)}{n_a * N_T}$	Penalidade por repetição de tipos de tarefas idênticos, onde: n_T , é o número de tarefas com tipos idênticos em um conjunto; n_a , é o número de atributos da tarefa.

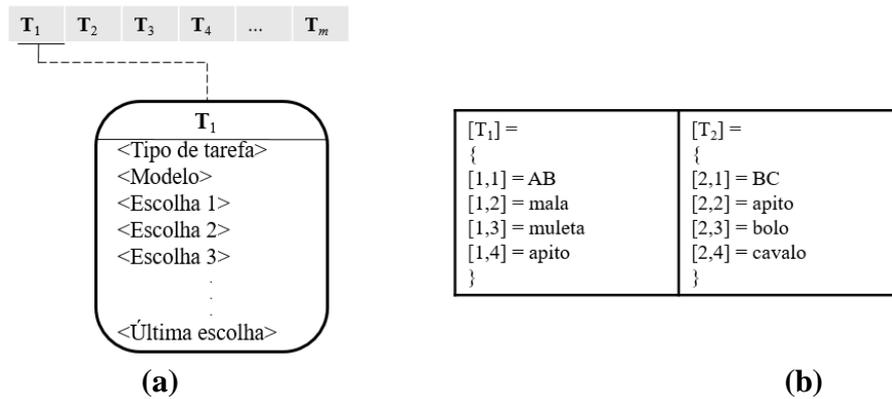
4.2.3. Algoritmos bio-inspirados

Neste subcapítulo, apresentam-se os aspectos referentes aos três algoritmos bio-inspirados implementados neste trabalho. Em especial, exemplos de codificação do algoritmo genético, algoritmo de otimização por enxame de partícula para variáveis discretas e algoritmo de otimização por colônia de formiga, são exibidos a seguir.

4.2.3.1. Algoritmo Genético para a geração de tarefas

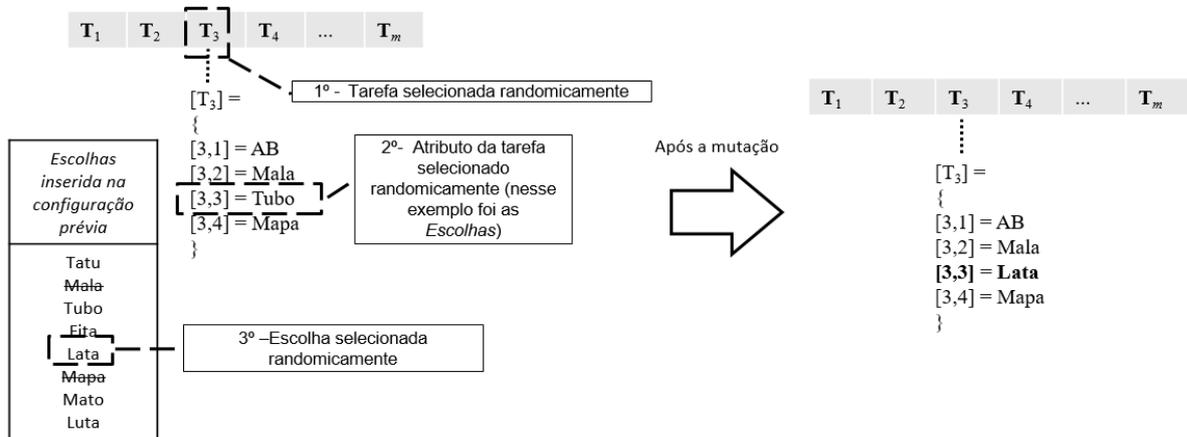
No problema de geração de tarefas com o GA, as soluções candidatas são modeladas para representar o conjunto de tarefas. Por sua vez, uma solução é composta dos atributos: tipo de tarefa; modelo; e escolhas. A Figura 4.20 ilustra a representação de uma solução com duas tarefas para o GA.

Figura 4.20 – (a) Estrutura de codificação do indivíduo e (b) exemplo de um indivíduo no GA.



No método de mutação proposto para o GA, cada campo da tarefa é selecionado aleatoriamente e substituído por outro parâmetro, como no exemplo da Figura 4.21. Esse método evita a repetição de escolhas em uma mesma tarefa bloqueando escolhas incompatíveis (como mostra a linha riscada em “Mala” e “Mapa” da Figura 4.21).

Figura 4.21 - Exemplo do método de mutação com seleção segundo a probabilidade de mutação.



A probabilidade de mutação e cruzamento no GA foi ajustada de acordo com a diversidade populacional; para esse objetivo, usamos a Equação 4.14 de auto adaptação (Zhu e Liu, 2004).

$$p' = \max \left(p_{min}, \min \left(\left(p_{max}, p \left(1 + \frac{\xi(d_t - d)}{d} \right) \right) \right) \right) \quad (4.14)$$

Onde:

p' : é a nova probabilidade para a próxima iteração do operador genético;

p : é a probabilidade atual do operador genético;

d : é a taxa de diversidade da população atual;

d_t : é o alvo da diversidade com valor 50% para manter uma população em equilíbrio;

ξ : é a constante de sensibilidade, 0,01 para cruzamento e 0,02 para controle de mutação);

p_{min} e p_{max} : são os limites mínimo e máximo de probabilidade dentro do intervalo de [0,1].

Após o processo evolutivo do GA, a melhor solução (ou seja, um conjunto de tarefas) é selecionada depois do critério de parada, para o instrutor que integra as tarefas em um software educacional.

4.2.3.2. Otimização por enxame de partículas para a geração de tarefas

No problema de geração de tarefas utilizando o ICPSO, a posição atual da partícula é um conjunto de distribuições de probabilidade, com um conjunto de probabilidades para cada atributo da tarefa, conforme apresentado no Capítulo 2 desta tese. A Figura 4.22 mostra duas tarefas com a representação de uma solução para este algoritmo. Para exemplificar, o atributo $y_{i,1,1}$ tem uma probabilidade de 0,5 de ser o tipo de tarefa AB - $P(\text{"AB"})$.

Figura 4.22 - Um exemplo de um indivíduo para a representação do ICPSO.

Tipos de tarefa	Modelos	Escolhas	Escolhas	Tipos de tarefa	Modelos	Escolhas	Escolhas
$[D_{i,1}] =$ { $d_{i,1,1}=P(\text{"AB"})=0.5$ $d_{i,1,2}=P(\text{"BC"})=0.2$ $d_{i,1,3}=P(\text{"CB"})=0.3$ }	$[D_{i,2}] =$ { $d_{i,2,1}=P(\text{"mala"})=0.4$ $d_{i,2,2}=P(\text{"apito"})=0.6$ }	$[D_{i,3}] =$ { $d_{i,3,1}=P(\text{"mala"})=0.2$ $d_{i,3,2}=P(\text{"apito"})=0.3$ $d_{i,3,3}=P(\text{"muleta"})=0.2$ $d_{i,3,4}=P(\text{"bolo"})=0.1$ $d_{i,3,5}=P(\text{"cavalo"})=0.2$ }	$[D_{i,4}] =$ { $d_{i,4,1}=P(\text{"mala"})=0.2$ $d_{i,4,2}=P(\text{"apito"})=0.4$ $d_{i,4,3}=P(\text{"muleta"})=0.1$ $d_{i,4,4}=P(\text{"bolo"})=0.2$ $d_{i,4,5}=P(\text{"cavalo"})=0.1$ }	$[D_{i,5}] =$ { $d_{i,5,1}=P(\text{"AB"})=0.5$ $d_{i,5,2}=P(\text{"BC"})=0.1$ $d_{i,5,3}=P(\text{"CB"})=0.4$ }	$[D_{i,6}] =$ { $d_{i,6,1}=P(\text{"mala"})=0.3$ $d_{i,6,2}=P(\text{"apito"})=0.7$ }	$[D_{i,7}] =$ { $d_{i,7,1}=P(\text{"mala"})=0.1$ $d_{i,7,2}=P(\text{"apito"})=0.1$ $d_{i,7,3}=P(\text{"muleta"})=0.2$ $d_{i,7,4}=P(\text{"bolo"})=0.5$ $d_{i,7,5}=P(\text{"cavalo"})=0.1$ }	$[D_{i,8}] =$ { $d_{i,8,1}=P(\text{"mala"})=0.3$ $d_{i,8,2}=P(\text{"apito"})=0.1$ $d_{i,8,3}=P(\text{"muleta"})=0.1$ $d_{i,8,4}=P(\text{"bolo"})=0.1$ $d_{i,8,5}=P(\text{"cavalo"})=0.4$ }
T_1				T_2			

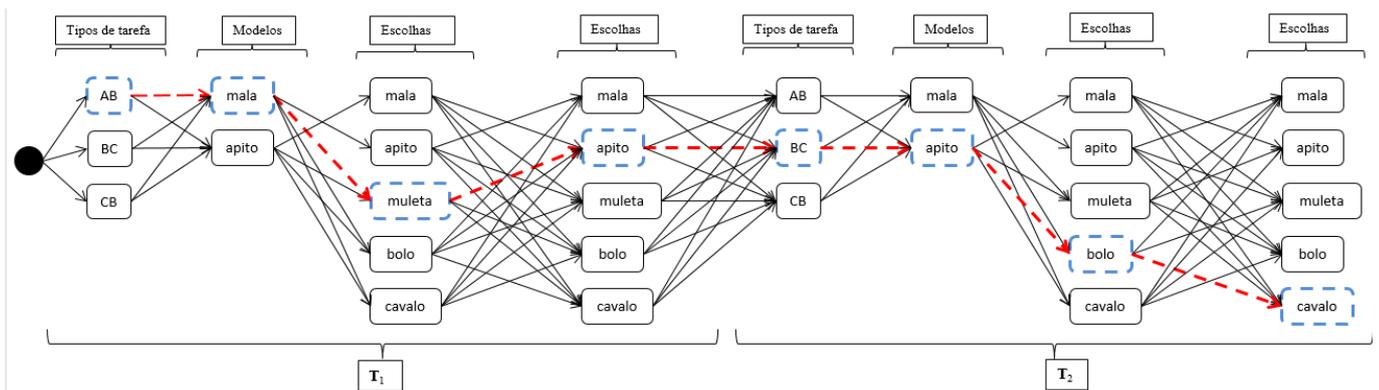
No ICPSO, a velocidade da partícula, a melhor posição de partícula e a melhor posição global do enxame têm representação semelhante à Figura 4.22. Para a criação da amostra de tarefas o método da roleta é usado para selecionar os atributos (tipos de tarefa e palavras). Para evitar a seleção das palavras repetidas em uma mesma tarefa, o seguinte critério é executado: se alguma palavra selecionada estiver na tarefa, a palavra será bloqueada e não será inserida como outra escolha da tarefa.

Os processos computacionais de atualização dos parâmetros e a definição de melhores vetores são semelhantes aos do PSO segundo Strasser *et al.* (2016), e desta forma o algoritmo produz um conjunto de tarefas para o tutor.

4.2.3.3. Otimização por colônia de formigas para a geração de tarefas

No problema de geração de tarefas usando ACO, cada campo do atributo (tipos de tarefas e palavras) correspondem a um nó no grafo, como mostra o exemplo na Figura 4.23.

Figura 4.23 - Um exemplo do grafo e do indivíduo (linha tracejada) no ACO.



A seleção de nós que são palavras está sujeita à restrição de que a mesma palavra não deve ser repetida para a mesma tarefa. Essa restrição é imposta bloqueando o caminho para outro nó através da atribuição de um valor igual a zero para a probabilidade de uma formiga selecionar este caminho. Após esta restrição ter sido aplicada, é necessário normalizar as probabilidades dentro de um intervalo de $[0, 1]$.

Após o processo iterativo, as formigas completam seus caminhos e é possível selecionar um conjunto de tarefas.

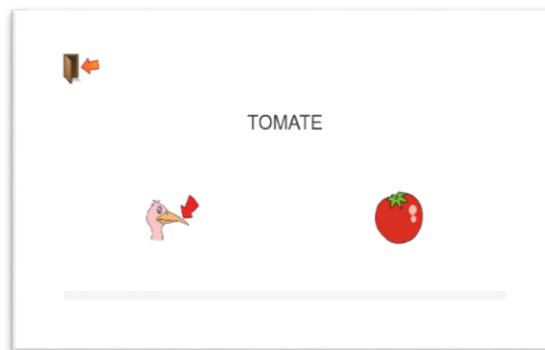
4.2.4. Softwares de aprendizado

Neste trabalho, foram desenvolvidos dois softwares de aprendizagem, como forma de implantar características interativas e lúdicas às tarefas de ensino, que podem ser baixados gratuitamente pelo link <http://linc.ufpa.br/amaru-mts/>. O primeiro software foi um aplicativo web, em execução em um navegador da internet, e o segundo software foi um jogo digital interativo. No Apêndice IV e Apêndice V pode-se visualizar mais detalhes sobre as ferramentas desenvolvidas.

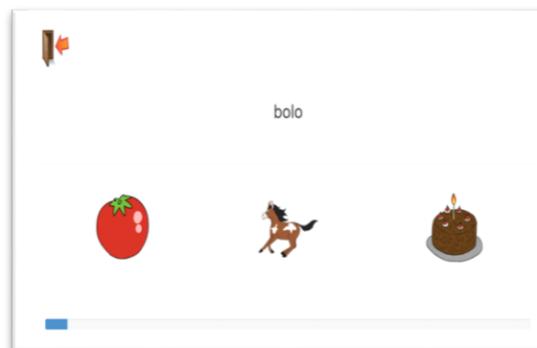
Nesses softwares os alunos podem tentar responder uma mesma tarefa em até três vezes. Se o aluno respondeu incorretamente por três vezes a mesma tarefa, o sistema pula a tarefa atual e apresenta a próxima tarefa. A interação entre o usuário e os softwares ocorrem através de interações com o mouse, com o objetivo de selecionar a resposta certa entre as opções exibidas na tela.

O aplicativo da web desenvolvido para este estudo, chama-se MTS-Player, que usa feedback visual e sonoro (reforços), como um indicador de perguntas respondidas corretamente ou incorretamente. Se o participante desejar interromper a sessão de treinamento, ele poderá usar um botão de saída localizado no canto superior esquerdo da tela. Uma barra de progresso mostra a progressão do aluno na sessão de ensino. A Figura 4.24 exhibe janelas do MTS-Player em execução.

Figura 4.24 - Aplicativo Web (MTS-Player) para o tipo de tarefa CB, com “TOMATE”(a) , e “bolo”(b) como modelos das tarefas.



(a)



(b)

O jogo digital desenvolvido para este estudo chama-se “As aventuras de Amaru”, que usa recursos como narrativa, personagens animados, “mini-games” e recompensas após a conclusão de desafios. O jogo apresenta a história de um alienígena chamado Amaru, que na trama do jogo, sofreu um acidente e caiu no planeta Terra com sua nave espacial. Para consertar a nave e retornar ao planeta natal, Amaru precisa aprender a se comunicar com os humanos, e

para isso ele conta com a ajuda de seu amigo robô no decorrer do jogo. Depois de apresentar o enredo do jogo, um pequeno tutorial é mostrado para o aluno, a fim de ajudar nos comandos básicos do software.

As tarefas são apresentadas como “mini-games” durante a progressão do jogo onde, o aluno deve selecionar a resposta correta, controlando o personagem principal com ações como, saltar, atirar projéteis e alcançar cubos flutuantes (Figura 4.25 (a)). O jogo distribui as tarefas em cinco fases e o aluno pode acompanhar seu progresso em um “mini-mapa” (Figura 4.25 (b)). Entre a apresentação de cada mini-game, o aluno pode coletar itens durante as fases do jogo, onde a pontuação e valor dos itens é proporcional ao número de tentativas corretas na tarefa anteriormente respondida. O *feedback* de cada tarefa (reforço positivo ou negativo) também é semelhante ao aplicativo web.

Figura 4.25 - Imagens das “Aventuras de Amaru”: (a) execução de um mini-game para o tipo de tarefa BC, (b) mini-mapa.



Para auxiliar o professor na inserção de tarefas, uma ferramenta foi desenvolvida para permitir a inclusão de arquivos CSV (*Comma-Separated Values*) no qual cada linha é uma tarefa e cada coluna é um atributo dessa tarefa, como mostrado na Figura 4.26. O jogo faz uma divisão uniforme das tarefas por estágio. Por exemplo, com um conjunto de 45 tarefas inseridas no jogo, cada fase terá 9 tarefas, seguindo a ordem estipulada no arquivo CSV.

Figura 4.26 -Arquivo de configuração CSV de um conjunto de tarefas.

1	Tipo de Tarefa	Escolhas	Resposta Correta	Treino ou Teste
2	BB	tatu luta lata	tatu	Treino
3	AB	lata fita tubo	lata	Treino
4	BC	mala mato luta	mala	Treino
5	CB	tatu lata mato	tatu	Treino
6	CB	mala tatu mapa	mala	Treino
7	CC	tatu tubo lata	tatu	Teste
8	BC	mala mapa mato	mala	Teste
9	AC	lata luta fita	lata	Treino
10	AC	lata luta fita	lata	Teste

As imagens, sons/áudios e textos foram feitos por um time especializado em arte multimídia e são guardados em uma pasta com suas respectivas configurações; isto possibilita a fácil inclusão e remoção dos itens usados para compor a tarefa. O gerador de tarefas produz o arquivo CSV com as tarefas geradas pelos algoritmos bio-inspirados. O arquivo em CSV e os arquivos multimídias podem ser especificados e direcionados a uma pasta presente no jogo e no aplicativo web, possibilitando praticidade e portabilidade dos recursos que serão utilizados para o ensino.

4.3. Considerações finais do capítulo

Neste capítulo foi apresentado o sistema proposto para a geração automática de tarefas de ensino. Para isso, os principais pontos foram detalhados com o objetivo de explicar o método de criação das tarefas adaptadas para o aluno levando em consideração a dificuldade das tarefas de ensino.

Esta tese buscou preencher a lacuna e solucionar limitações com base nos trabalhos correlatos. Dentre eles estão:

- O preenchimento da lacuna na criação de tarefas para o procedimento MTS por um novo método para geração das tarefas de modo automático com algoritmos bio-inspirados.
- Ajuste da dificuldade das tarefas de ensino adaptadas ao aluno utilizando dois fatores de dificuldade de uma tarefa de ensino, o fator da similaridade entre escolhas e o fator da dificuldade dos tipos de tarefas.
- Implementação de softwares interativos de aprendizagem para melhorar a aceitabilidade e interatividade por parte dos alunos diante das tarefas de ensino.

O próximo capítulo apresentará os diversos experimentos e resultados desta tese. Estes experimentos se baseiam no sistema proposto no presente capítulo, propondo validar os algoritmos bio-inspirados implementados, e também aplicar as tarefas geradas pelos mecanismos de inteligência computacional desenvolvidos.

5. Experimentos e Resultados

5.1. Considerações iniciais do capítulo

Após a implementação dos algoritmos bio-inspirados, dos programas auxiliares e dos softwares de aprendizagem, este trabalho propôs a validação dos métodos desta tese realizando experimentos e estudos de caso. Estas validações buscaram analisar tanto os aspectos computacionais quanto a validação com agentes humanos que interagiram com o sistema de geração de tarefas.

Desta forma, os experimentos apresentados neste capítulo buscam revelar aspectos do uso do sistema para alunos em fase inicial de aprendizagem, mais precisamente para o ensino básico de leitura.

5.2. Síntese dos experimentos e validações realizadas

O primeiro experimento consistiu em uma comparação computacional dos algoritmos bio-inspirados. Buscou-se executar todos os algoritmos modelados, com base em perfis de desempenho do aprendizado de alunos, para verificar qual algoritmo seria o mais estável em relação à convergência do *fitness* para este domínio de aplicação.

O segundo experimento buscou verificar se as tarefas geradas seriam adequadas, em termos de dificuldade, para serem apresentadas a crianças entre 5 e 7 anos e no início da alfabetização. Nesta validação buscou-se a opinião e a experiência de profissionais da análise comportamental e pedagogia. Essa apreciação com especialistas foi importante para verificar o correto direcionamento dos resultados, previamente à aplicação para a sala de aula com alunos reais.

A terceira validação contou com o ensino individualizado e buscou verificar se a abordagem para a geração de tarefas poderia ser utilizada com alunos. Além disso, indicadores relativos ao aprendizado e o engajamento foram coletados através de testes e entrevistas; de modo a melhor indicar se a abordagem implementada está gerando tarefas adequadas para crianças em um processo de ensino maior e mais detalhado.

Por último, a validação sobre o engajamento foi realizada por meio de entrevistas aplicadas a grupos de alunos. Os softwares de aprendizado (jogo digital e aplicativo web) foram usados para apresentar tarefas adaptadas para os alunos. Estes softwares são interfaces de tarefas e podem ser fundamentais para o ensino, mostrando aspectos de engajamento necessário

para um aprendizado mais interessante para os alunos, como sugere algumas pesquisas apresentadas nesta tese.

Destaca-se que o projeto dos experimentos realizados com seres humanos foi submetido à Plataforma Brasil e ao Comitê de Ética e Pesquisa (CEP) que aprovaram o presente estudo (Anexo A). Nos subcapítulos que seguem, os processos e configurações dos experimentos são expostos, e os resultados são apresentados.

5.3. Validação computacional dos algoritmos bio-inspirados

Neste trabalho, um experimento foi realizado para avaliar a eficácia dos algoritmos bio-inspirados (GA, ICPSO, ACO) para a geração automática de conjuntos adaptados de tarefas usando o vocabulário brasileiro-português.

Para esse objetivo, foram usados pré-testes com respostas que simulam perfis de aprendizagem de alunos, determinados como: (1) desempenho inicial, com cerca de 25% das respostas corretas distribuídas na primeira parte das tarefas respondidas; e (2) desempenho intermediário, com aproximadamente 50% das respostas corretas distribuídas na primeira parte das tarefas respondidas.

Os pré-testes foram divididos em seis blocos de tarefas com seis tarefas para cada tipo. Os modelos e escolhas foram fixados enquanto apenas os tipos de tarefas foram alterados. O padrão de *string* para definir uma tarefa de pré-teste, foi o seguinte: [<tipo de tarefa>, <modelo>, <escolha 1>, <escolha 2>]. Esta estrutura de pré-teste é mostrada abaixo:

- 1ª tarefa até a 6ª tarefa = [BB; mala; tatu; tubo; luta];
- 7ª tarefa até a 12ª tarefa = [CC; mala; tatu; tubo; luta];
- 13ª tarefa até a 18ª tarefa = [AB; tatu; mala; tubo; luta];
- 19ª tarefa até a 24ª tarefa = [CB; tatu; mala; tubo; luta];
- 25ª tarefa até a 30ª tarefa = [BC; tubo; luta; mala; tatu];
- 31ª tarefa até a 36ª tarefa = [AC; tubo; luta; mala; tatu].

O conjunto inicial de dados para a configuração para a geração de tarefas está descrito na Tabela 5.7 e o fluxo de dificuldade utilizado foi em formato de “degraus de escada”, semelhante à Figura 4.17 presente no Capítulo 4.

Tabela 5.7 – Configuração para a geração de tarefas.

Perfil de tarefas respondidas	Modelos	Escolhas	Número mínimo e máximo de palavras por tarefas	Número de tarefas a ser geradas	Tipos de tarefas	Tamanho da solução (cromossomo, partícula, caminho da formiga)
Desempenho inicial	Tatu Mala Tubo	Tatu	3	36	BB, CC, AB	144
		Mala				
Desempenho Intermediário		Tubo		54	CC, AB, CB, BC, AC	216
		Fita				
		Lata				
		Mapa				
		Mato				
		Luta				

Posteriormente a inclusão das configurações para a geração de tarefas, os parâmetros dos algoritmos foram ajustados após diversos testes, conforme apresentado na Tabela 5.8. Todos os algoritmos usaram elitismo, com o número de iterações e o número de indivíduos configurados tendo os valores de 500 e 50, respectivamente. Neste experimento, cada algoritmo foi executado 100 vezes. O número de iterações e indivíduos foi padronizado para existir uma igualdade no número de execuções da função de *fitness* entre os algoritmos, permitindo uma avaliação justa dos resultados.

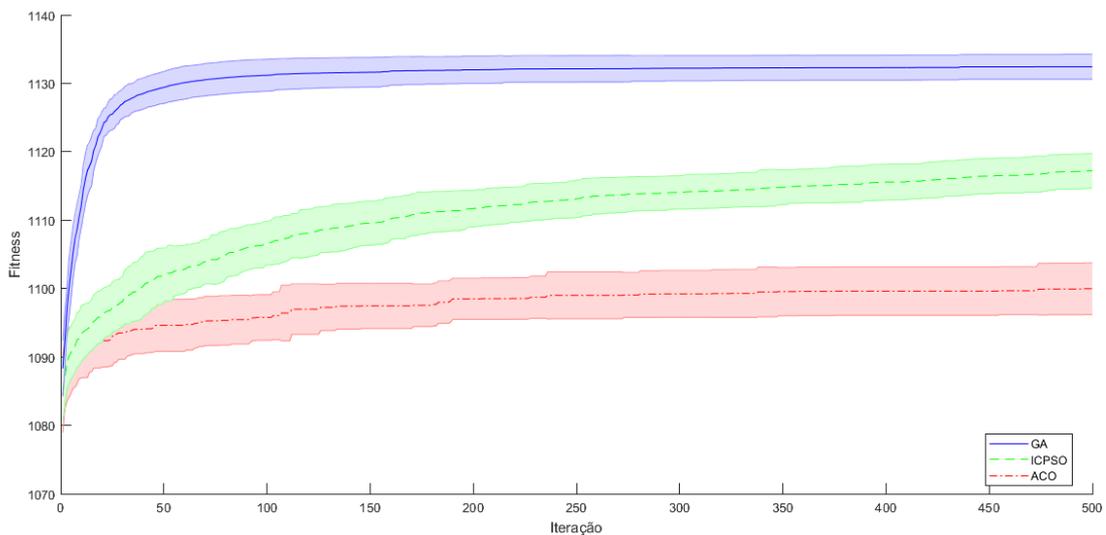
Tabela 5.8 – Configuração do GA, ICPSO e ACO.

Algoritmo	Nome do método ou parâmetro	Descrição ou valor do parâmetro
GA	Método de seleção	Torneio
	Método de cruzamento	Uniforme (Abuiziah & Shakarneh, 2014)
	Probabilidade de cruzamento	Auto-adaptativo conforme subcapítulo 4.2.3.1
	Método de mutação	Semi-aleatório conforme subcapítulo 4.2.3.1
	Probabilidade de mutação	Auto-adaptativo conforme subcapítulo 4.2.3.1
ICPSO	Coefficiente cognitivo c_1	2.5
	Coefficiente social c_2	1.5
	Peso de inercia w	0.729
	Fator de escala ϵ	0.6
ACO	Evaporação do feromônio ρ	0.2

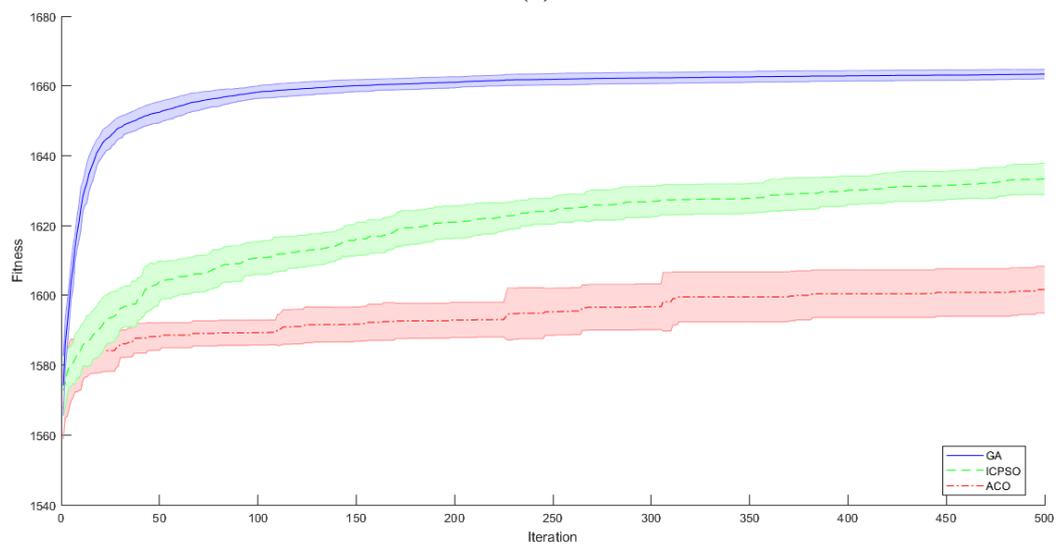
O fator de dificuldade de tipo de tarefa (βt_{κ}), exibido na Equação 3.9, são os fatores de dificuldade do tipo de tarefa definidos por especialistas presentes na equipe de desenvolvimento deste trabalho, que assumiram os seguintes valores: $\beta t_{BB} = -3$, $\beta t_{CC} = -2.64$, $\beta t_{AB} = 0.07$, $\beta t_{CB} = 1.02$, $\beta t_{BC} = 2$, $\beta t_{AC} = 3$, respectivamente. Estes valores foram normalizados em um intervalo de $[-3, +3]$, que são limites representativos apropriados para serem utilizados na Equação 3.9 que produz a dificuldade de uma tarefa. Valores menores para βt_{κ} representam uma tarefa mais fácil e valores maiores representam um grau de dificuldade maior.

A Figura 5.27 mostra as curvas de *fitness* para todos os algoritmos para os dois perfis de desempenho. O GA apresentou uma taxa de convergência maior que os outros algoritmos. A convergência do ICPSO foi lenta quando comparada ao GA, embora vários testes tenham sido realizados para definir os parâmetros e, assim, tentar acelerar a convergência. O ACO converge para um estado estável, mas possui um valor de *fitness* menor que os outros algoritmos.

Figura 5.27 -Comportamento da curva de convergência de *fitness*. A linha central é a média e o sombreado é o desvio padrão dos melhores indivíduos. (a) desempenho inicial, (b) desempenho intermediário.



(a)

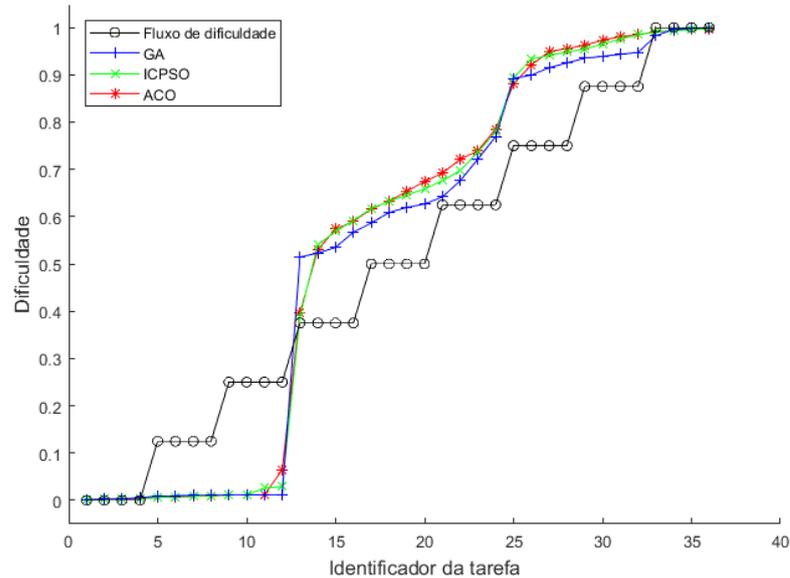


(b)

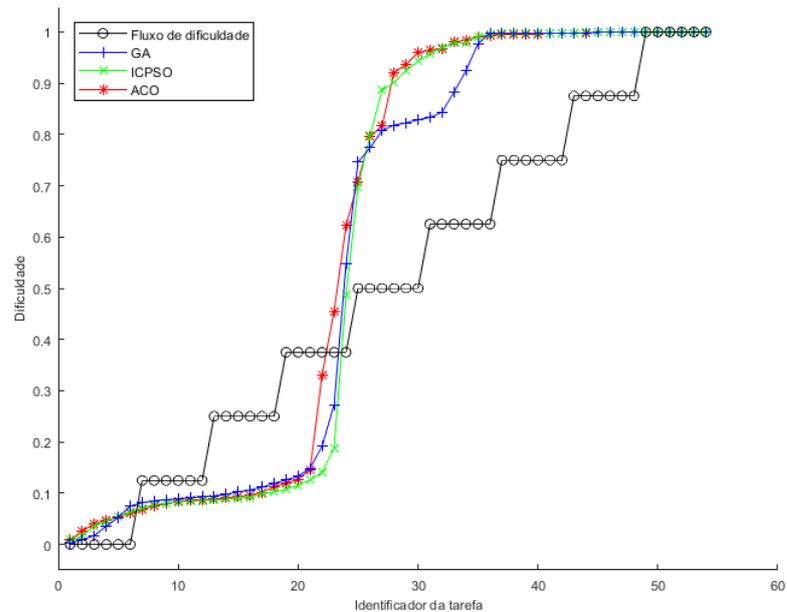
A Figura 5.28 mostra a média de dificuldade para o melhor conjunto de tarefas obtidas nas 100 execuções. Observa-se, por exemplo, que houve um salto na curva entre as tarefas 20 e 30 para o perfil de desempenho intermediário. Isto é uma característica natural visto que neste intervalo específico existiam poucas tarefas com valores de dificuldade validos, provavelmente

devido pouca variabilidade de palavras e tipos de tarefas inseridos na configuração inicial de geração de tarefas. O perfil de desempenho intermediário, por representar um aluno com maior conhecimento que o perfil de desempenho inicial, pode ter interferido nesta característica, limitando a quantidade de tarefas possíveis neste intervalo específico. A Tabela 5.9 e Tabela 5.10 mostram uma amostra dos melhores conjuntos de tarefas geradas pelo GA.

Figura 5.28 -Dificuldade de tarefas geradas pelo perfil de desempenho (a) inicial e (b) intermediário.



(a)



(b)

Tabela 5.9 – Uma amostra de tarefas geradas em ordem de apresentação para os alunos para o perfil de desempenho inicial.

Conjunto de tarefas gerados para o perfil de desempenho inicial (36 tarefas)			
Tipo de tarefa	Modelo	Escolha1	Escolha2
BB	tubo	tatu	fita
BB	tatu	mapa	lata
BB	tubo	luta	fita
(...)			
AB	mala	tatu	fita
AB	tatu	tubo	mala
AB	tatu	mala	tubo
(...)			
CC	mala	mapa	mato
CC	mala	mapa	mato
CC	tubo	luta	fita

Tabela 5.10 – Uma amostra de tarefas geradas em ordem de apresentação para os alunos para o perfil de desempenho intermediário.

Conjunto de tarefas gerados para o perfil de desempenho intermediário (54 tarefas)			
Tipo de tarefa	Modelo	Escolha1	Escolha2
AB	tatu	mato	lata
AB	mala	mato	mapa
AB	tubo	tubo	fita
(...)			
CB	tatu	lata	luta
CB	tubo	fita	lata
CB	tubo	mala	tatu
(...)			
BC	tatu	lata	fita
BC	mala	tatu	lata
BC	mala	lata	fita

Adicionalmente, os erros de predição foram calculados por meio do RMSE (*Root mean Square Error* – erro da raiz quadrada média), entre a dificuldade média das tarefas geradas e o fluxo de dificuldade. Define-se como erro, a distância das dificuldades das tarefas geradas em relação ao fluxo de dificuldade, onde quanto menor a distância, menor o erro. No perfil de desempenho inicial, o RMSE para GA, ICPSO e ACO foi 0,1274, 0,1362 e 0,1371, respectivamente. Para o perfil de desempenho intermediário, o RMSE para GA, ICPSO e ACO foi 0,1876, 0,2151 e 0,2123, respectivamente.

Neste estudo não buscamos testar os algoritmos bio-inspirados para um perfil de aluno com desempenho avançado, pois o intuito deste experimento foi promover o aprendizado de alunos que não sabiam ler ou tinham alguma dificuldade no processo de aprendizagem.

O melhor desempenho do GA pode ser atribuído a seu aspecto evolucionário, contudo destaca-se que o algoritmo ICPSO mostrou que poderia se aproximar do GA se fosse utilizado

um maior número de iterações, como pode-se visualizar na curva de *fitness*. Com os resultados deste experimento, pode-se inclusive recomendar um número de iterações menor para o GA (200 por exemplo), de forma que poderia-se gerar tarefas mais rapidamente. Assim, observou-se que o GA mostrou rapidez em convergir indicando boa estabilidade e gerando tarefas mais próximas ao fluxo de dificuldade do que os demais algoritmos.

5.4. Validação da dificuldade das tarefas por especialistas

Antes de utilizar as tarefas geradas para serem aplicadas com alunos em fase de aprendizagem, uma validação com especialistas foi realizada com o objetivo de verificar se as dificuldades das tarefas geradas são adequadas para as crianças que estão iniciando o aprendizado na alfabetização. Para isso, foi utilizado um questionário e opiniões de especialistas foram coletadas.

Cada questão do questionário foi composta por uma tarefa de ensino exibida em uma imagem e 5 opções inspirado na escala *Likert* (William M K Trochim, 2006). O questionário possui 16 questões com o seguinte padrão: “Qual é o nível de dificuldade da tarefa de ensino para crianças em fase de aprendizado na leitura? : 1 - Muito fácil, 2 - Fácil, 3 - Média, 4 - Difícil, 5 - Muito difícil”.

Nove especialistas foram consultados com experiência de 1 a 4 anos de atuação em estudos de alfabetização. Os especialistas consultados foram professores, psicólogos, pesquisadores na área da alfabetização e pedagogos. As respostas dos questionários foram normalizadas entre 0 e 1, permanecendo na mesma escala de valores para dificuldades geradas pelo algoritmo proposto nessa pesquisa.

Neste contexto, usando a correlação de Pearson foi possível analisar a relação entre a média da opinião dos especialistas e os valores gerados das dificuldades. O resultado revela valor de R igual a 0,9037, que indica uma forte correlação entre a opinião dos especialistas e a dificuldade das tarefas geradas. Essa forte correlação, de acordo com os especialistas, mostra que as tarefas de ensino geradas pelo algoritmo são adequadas para crianças em estágio inicial de alfabetização.

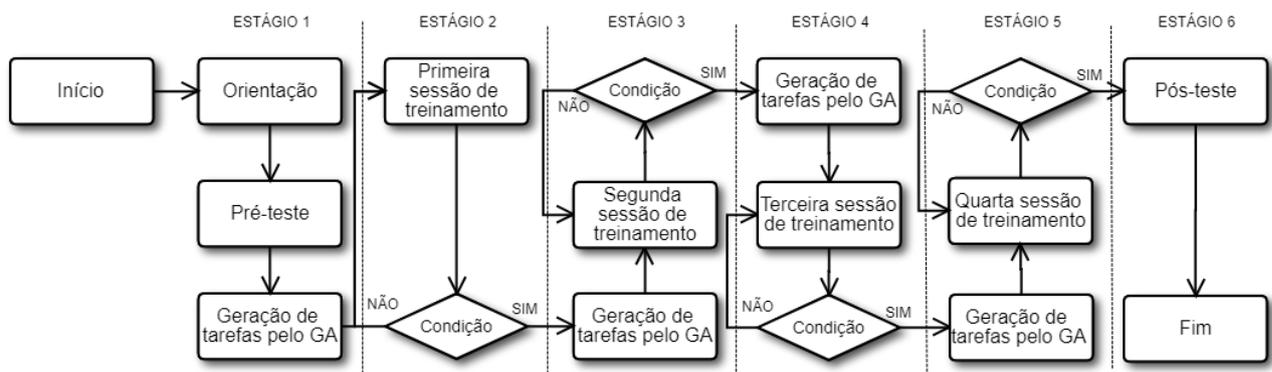
5.5. Validação das tarefas geradas para o processo de ensino individualizado

Uma atividade foi concebida como estudo de caso envolvendo a aplicação das tarefas geradas para os alunos no estágio de desenvolvimento da alfabetização. O objetivo foi mostrar

a aplicabilidade de utilizar algoritmos bio-inspirados como forma de automatizar o processo, e analisar se as tarefas geradas tiveram algum efeito positivo no processo de aprendizagem. Por conseguinte, analisou-se o número de respostas corretas antes e depois da aplicação do conjunto de tarefas geradas e analisou-se o engajamento dos participantes.

A Figura 5.29 mostra o desenho da atividade aplicada aos alunos. Na primeira etapa, foi dada orientação sobre como utilizar os softwares de aprendizagem que exibem as tarefas no computador. As tarefas respondidas no pré-teste foram usadas como entrada para gerar o conjunto de tarefas pelo GA. Nos estágios 2 a 5, os participantes foram treinados com conjuntos de tarefas gerados pelo GA, alternando os tipos de tarefas. As respostas da primeira sessão de treinamento foram usadas para gerar o conjunto de tarefas para a segunda sessão, e assim por diante, até a última sessão de treinamento ser concluída. Cada estágio tem uma condição e, se a condição for cumprida, o aluno passa para o próximo estágio; caso contrário, o aluno deve repetir a sessão de treinamento. Antes de passar para a próxima etapa, o aluno deve responder corretamente a pelo menos 60% das tarefas. A porcentagem de 60% foi adquirida empiricamente por consulta a especialistas em psicologia presentes na equipe deste trabalho.

Figura 5.29 – Desenho da atividade individualizada.



Na etapa final, uma entrevista semi-estruturada foi realizada por profissionais da educação, com perguntas elaboradas para investigar o nível de dificuldade dos comandos de entrada e também o engajamento do aluno. Como os participantes eram crianças, foram realizadas perguntas simples, aplicando a liberdade de uma entrevista semi-estruturada (Irvine, Drew, & Sainsbury, 2013).

As seguintes perguntas foram feitas: (1) “Você gostou da atividade?”; (2), “Você gostaria de fazer a atividade novamente?”; (3), “Você foi capaz de fazer a atividade?”; (4), “Quais foram os comandos de teclado e mouse que você usou na atividade? (resposta positiva, se o aluno foi capaz de mostrar os comandos corretamente) ”; (5), “Você teve alguma

dificuldade em fazer esta atividade?”. Com base nos dados coletados, as respostas foram categorizadas como “Sim” (resposta positiva) ou “Não” (resposta negativa), observando os comentários, o modo de falar, as expressões faciais e anotando as respostas. Por exemplo: uma criança balançando a cabeça verticalmente confirma um “sim” (resposta positiva); as expressões “legal”, “bom” e “bonito” seriam classificadas como “sim” (resposta positiva); e as expressões “ruim”, “bobo” e “feio” foram classificadas como “não” (resposta negativa).

Esta atividade foi aplicada em um laboratório de informática de uma escola pública de da cidade de Belém, Estado do Pará, Brasil. Uma turma de alunos com um total de 18 alunos, com idades entre 5 e 6 anos, foi convidada a participar. Dentre os participantes, 5 alunos abandonaram a atividade, 4 alunos não concluíram a etapa de treinamento e 3 perderam pelo menos um dia de atividade, restando 6 alunos que completaram a atividade. A execução deste experimento com 6 alunos visou identificar se existe uma aplicabilidade na utilização da nossa abordagem, ou seja, meramente verificar se é possível gerar tarefas e aplica-las em ambientes reais de aprendizado. Assim, este experimento possui um caráter de validação e estudo de caso, não buscando indicar se nosso método é eficaz em comparação com outros métodos de aprendizado, visto que para esta constatação seria necessário um maior número de alunos, maior número de especialistas educacionais para auxiliar a aplicação desta abordagem e métodos de aprendizado comparativos vindouros da psicologia educacional.

Para a aplicação deste estudo de caso, o aluno precisou concluir a atividade e, por esse motivo, fora decidido considerar apenas os dados dos seis alunos que participaram de toda a atividade. Os participantes foram submetidos individualmente a 6 etapas de ensino de 30 a 50 minutos, completando um total de 2 meses de atividades. Cada aluno respondeu pelo menos 648 tarefas em todo o processo. A equipe supervisora da escola e os responsáveis legais do aluno assinaram o termo de consentimento (Apêndice III) autorizando esse estudo. A Figura 5.30 mostra 2 crianças em atividade individual de ensino.

Figura 5.30 – Atividade de ensino individual para dois alunos.

O primeiro resultado da atividade veio das entrevistas semi-estruturadas, compiladas na Tabela 5.11. Em uma análise geral, verificou-se que a maioria dos alunos aceitou a atividade de forma positiva. Além disso, os alunos foram capazes de fazer a atividade e interagir corretamente com o computador.

Tabela 5.11 – Resultado da entrevista semi-estruturada.

Perguntas	SIM	NÃO
1. Você gostou da atividade?	83,33%	16,66%
2. Você gostaria de fazer a atividade novamente?	100%	0
3. Você conseguiu fazer a atividade?	100%	0
4. Quais foram os comandos de teclado e mouse usados na atividade? (SIM, se o aluno foi capaz de mostrar os comandos corretamente)	100%	0
5. Você teve alguma dificuldade em fazer esta atividade?	33,33%	66,66%

Outro resultado foi em relação ao número de repetições das etapas do treinamento. Este dado foi coletado pois, um grande número de repetições pode desencorajar os alunos e sugerir que as tarefas geradas não são apropriadas para eles. A Tabela 5.12 exibe o número de repetições nas etapas de treinamento, com médias entre 0,5 e 0,66, indicando que as tarefas geradas foram aceitáveis para esses alunos.

Tabela 5.12 – Número de repetições de estágios.

	Estágio 2	Estágio 3	Estágio 4	Estágio 5
Aluno 1	0	0	0	1
Aluno 2	2	0	1	0
Aluno 3	1	0	0	1
Aluno 4	0	1	1	0
Aluno 5	0	1	0	1
Aluno 6	0	2	2	0
Média	0.5	0.66	0.66	0.5

O terceiro resultado baseia-se nas respostas corretas durante a atividade, mostradas na Tabela 5.13. Os participantes obtiveram um melhor desempenho no pós-teste do que no pré-teste, indicando um efeito positivo no aprendizado desses alunos.

Tabela 5.13 – Número geral de respostas corretas do pré-teste e pós-teste (número total de 36 tarefas para cada aluno).

	Pré-teste	Pós-teste
Percentagem média de respostas corretas	53.24%	78.65%
Desvio padrão de respostas corretas	34.29%	16.05%

5.6. Engajamento dos softwares de aprendizagem

Um outro estudo de caso foi realizado a fim de identificar o engajamento dos alunos utilizando os softwares de aprendizado desenvolvidos ao mesmo tempo que utilizam as tarefas de ensino geradas. Desta forma, o jogo digital “As aventuras de Amaru” e o aplicativo web “MTS-Player” foram objetos desse estudo. 22 participantes foram convidados com idades entre 6 e 7 anos (1ª série) e divididas aleatoriamente em 4 grupos: 5 crianças no “Grupo A: Web-Game”, 6 no “Grupo B: Game-Web”, 4 no “Grupo C: Web-Web” e 3 no “Grupo D: Game-Game”. 4 participantes faltaram a atividade em pelo menos um dos dias e por esse motivo os dados coletados desses participantes não adentraram na análise do estudo. Da mesma forma como o experimento anterior, este experimento utilizou dados de 18 alunos e buscou mostrar a aplicabilidade da utilização das tarefas geradas, mas neste caso, com foco no engajamento e ludicidade que os softwares desenvolvidos poderiam causar.

Os participantes realizaram 2 sessões de 30 a 50 minutos cada. O estudo foi conduzido em um laboratório de informática da Universidade Federal do Pará, Belém, Brasil. A Figura 5.31 mostra uma sessão da atividade de ensino.

Figura 5.31 – Aplicação da sessão de ensino em um laboratório de computação.



O estudo para a identificação do engajamento foi conduzido conforme a Figura 5.32. No primeiro momento do primeiro dia, os participantes tiveram uma pequena orientação sobre o uso de aplicações computacionais; e no segundo momento do primeiro dia, os participantes foram apresentados às tarefas. Em um terceiro momento, a entrevista semi-estruturada foi aplicada individualmente. Esta estrutura foi replicada de maneira semelhante no segundo dia.

Os conjuntos de “tarefas 1” e “tarefas 2” foram tarefas diferentes e foram realizadas no primeiro e segundo dia, respectivamente.

Figura 5.32 – Desenho da atividade para a turma para a coleta do engajamento dos alunos.

Grupos	Primeiro momento do primeiro dia	Segundo momento do primeiro dia	Terceiro momento do primeiro dia	Primeiro momento do segundo dia	Segundo momento do segundo dia	Terceiro momento do segundo dia
Grupo A: Web-Game	Orientação	Execução do conjunto de tarefas 1 , com o aplicativo web.	Execução da entrevista semi-estruturada pela primeira vez.	Orientação	Execução do conjunto de tarefas 2 , com o jogo digital.	Execução da entrevista semi-estruturada pela segunda vez.
Grupo B: Game-Web		Execução do conjunto de tarefas 1 , com o jogo digital.			Execução do conjunto de tarefas 2 , com o aplicativo web	
Grupo C: Web-Web		Execução do conjunto de tarefas 1 , com o aplicativo web			Execução do conjunto de tarefas 2 , com o aplicativo web	
Grupo D: Game-Game		Execução do conjunto de tarefas 1 , com o jogo digital.			Execução do conjunto de tarefas 2 , com o jogo digital.	

As respostas das entrevistas foram categorizadas em três respostas para cada questão: “Sim” (resposta positiva), “Não” (resposta negativa) e “sem resposta” (resposta neutra), similar ao estudo do subcapítulo 5.5. As respostas foram compiladas na Tabela 5.14.

Tabela 5.14 – Porcentagem de respostas positivas nos dois dias do experimento.

Perguntas	Grupo A Web-Game		Grupo B Game-Web		Grupo C Web-Web		Grupo D Game-Game	
	Primeiro dia	Segundo dia	Primeiro dia	Segundo dia	Primeiro dia	Segundo dia	Primeiro dia	Segundo dia
1. Você gostou da atividade?	100%	100%	100%	100%	100%	100%	100%	100%
2. Você gostaria de fazer a atividade novamente?	0%	20%	83%	83%	75%	50%	33%	33%
3. Quais foram os comandos que você usou na atividade? (Sim, se o aluno puder usar os comandos do teclado e do mouse corretamente)	100%	80%	100%	100%	50%	75%	66%	100%
4. Você conseguiu fazer a atividade?	80%	100%	100%	100%	100%	100%	100%	100%

Na questão 1, observou-se que todos os quatro grupos responderam positivamente. A questão número 2, houve um resultado heterogêneo: apenas o “Grupo B Game-Web” obteve uma resposta positiva com mais de 80%; seguido pelo “Grupo C Web-Web” com mais da metade de respostas positivas; pelo “Grupo D Game-Game” com menos da metade; e, finalmente, pelo “Grupo A Web-Game” com os menores percentuais. Os baixos percentuais da pergunta número 2 talvez se devam ao processo de adaptação no primeiro contato dos alunos

com o aplicativo “Game” e “Web”, embora o “Game” pudesse aumentar com sucesso os resultados no caso do Grupo A, devido a característica de ludicidade.

Em relação à questão número 3, a maioria dos grupos respondeu positivamente e que poderiam usar os comandos do teclado e do mouse corretamente. Mesmo com a pequena orientação no início da atividade, os alunos demonstraram ter conhecimento sobre os dispositivos. Em consequência da questão número 3, a maioria dos grupos foi capaz de realizar a atividade, de acordo com a questão número 4.

Outra questão foi aplicada apenas ao “Grupo A: Web-Game” e “Grupo B: Game-Web”, para coletar a preferência em relação aos softwares de aprendizado: “Qual software você prefere? ”. Em relação à preferência entre os softwares de aprendizado, o “Grupo A: Web-Game” teve um total de 83%, 17% não responderam à questão. O “Grupo B: Game-Web” teve um total de 50% que preferem o jogo, 17% não preferem o aplicativo web e 33% não responderam à questão.

Este resultado mostra uma preferência pelo jogo digital, porém ambas as ferramentas se mostram acessíveis a essa turma. Esse resultado também indica que, mesmo sem treinamento prévio, a exposição contínua a conjuntos de tarefas é adequada para os participantes aprenderem como lidar com as ferramentas.

5.7. Considerações finais do capítulo

A implementação destes experimentos foi certamente um desafio. O sucesso das aplicações demonstrou também que existe viabilidade na execução dos algoritmos em ambientes de ensino. Mesmo assim, ainda existe uma quantidade razoável de experimentos, tanto computacionais como comportamentais, que podem ser realizados visando melhorar a abordagem de geração de tarefas. Estes experimentos e implementações são elencados como pesquisas futuras ao final da tese. No próximo capítulo são discutidos os resultados desta tese concluindo a presente pesquisa.

6. Conclusão

De acordo com a Avaliação Nacional de Alfabetização, em 2016, mais de 50% das crianças entre 5 e 8 anos tinham conhecimento insuficiente na leitura e escrita. A região Norte do Brasil apresentou o dado mais preocupante, com aproximadamente 70% das crianças em idade escolar apresentando alguma dificuldade em ler e escrever (Bermúdez, 2017). Alguns programas de ensino buscam auxiliar na mitigação deste problema auxiliando no ensino em áreas com baixo aproveitamento no aprendizado. Contudo, esses programas são dependentes de um grande esforço manual para a criação de tarefas individualizadas.

Esta tese apresentou uma abordagem baseada em meta-heurísticas bio-inspiradas para a geração de tarefas adaptadas utilizando o procedimento *Matching-to-Sample*. Os experimentos mostraram que os algoritmos bio-inspirados podem gerar um conjunto de tarefas com dificuldades adaptadas obedecendo ao fluxo de ensino inserido pelo tutor. Três meta-heurísticas (GA, ICPSO e ACO) foram comparadas, onde o GA apresentou melhor taxa de convergência, observando-se também que o grau de dificuldade das tarefas geradas permaneceu próximo ao fluxo de dificuldade previamente selecionado.

A opinião dos especialistas em relação à dificuldade das tarefas geradas mostrou correlação positiva, indicando que essas tarefas poderiam ser usadas para o ensino de crianças em fase inicial de alfabetização.

Após a aplicação de uma atividade de ensino, foi possível analisar se as tarefas geradas podem ser utilizadas em ambientes reais de aprendizado. Os resultados indicaram que houve uma melhoria na aprendizagem das crianças nos estágios iniciais do desenvolvimento da alfabetização. A análise das entrevistas semi-estruturadas mostrou que as tarefas geradas foram aceitas e não houve problemas no entendimento do uso de comandos ou dispositivos computacionais que pudessem afetar o processo de ensino. Assim, é possível utilizar o GA como uma meta-heurística para gerar uma quantidade considerável de tarefas, justificando seu uso com um número maior de alunos e com experimentos maiores ao molde da análise comportamental.

Este trabalho também apresentou um estudo sobre softwares de aprendizagem e interativos que apresentam as tarefas adaptadas para alunos. A utilização de um aplicativo web (MTS-Player) e um jogo digital (As Aventuras de Amaru) demonstraram a possibilidade de

utilizar uma sequência organizada de tarefas. Contudo, o resultado da atividade em grupo apresentou uma indicação que as tarefas MTS em um ambiente lúdico (jogo digital) podem engajar mais os alunos em fase inicial de aprendizagem. No entanto, para os estudantes com dificuldades na operação dos controles e comandos com o teclado e o mouse, é recomendável o uso do aplicativo web, pois o software possui uma interface mais simples do que o jogo digital.

O diferencial dos softwares de aprendizado lúdicos para outros softwares informatizados projetados para o ensino de habilidades de leitura, está nas características de integrar o procedimento de ensino em uma narrativa junto com uma boa interatividade. O jogo digital apresenta tarefas MTS como uma série de desafios contínuos, implementados como mini-games, planejados para entreter e ensinar ao mesmo tempo. Os itens e estímulos deste jogo digital são integrados à história, fazendo sentido dentro da narrativa, e assim evitando que o aluno se distraia durante o jogo. Assim, uma vantagem é que os softwares de aprendizado podem ser usados em atividades escolares ou fora da escola (por exemplo, *homeschooling*), aumentando as possibilidades de uso.

Com base no exposto, estima-se dizer que o objetivo principal foi completado, ou seja, a geração de tarefas com dificuldade adaptadas para crianças em fase inicial de alfabetização através de algoritmos bio-inspirados; promovendo as seguintes contribuições para a área de estudo:

- Viabilização da geração de um conjunto completo de tarefas de ensino para o procedimento MTS seguindo um fluxo de ensino.
- Viabilização de uma abordagem que engloba a aplicação de um pré-teste, a configuração de ensino, a geração de tarefas e o ensino em softwares de aprendizagem com as tarefas geradas; sendo útil sua utilização em ambientes reais de ensino como salas informatizadas de aula.
- Propõe um modelo que estima a dificuldade de uma tarefa do procedimento MTS e adaptada ao aluno, através da Equação 4.9.
- Disponibiliza a modelagem de um algoritmo bio-inspirado (algoritmo genético) apto para a geração de tarefas adaptadas e obedecendo a configuração de ensino ajustada por um especialista na área da análise comportamental.
- Permite o ensino individualizado, que respeita o modo de aprendizado de cada aluno, viabilizando um melhor ensino durante o processo de alfabetização.
- Dispõe o resultado de um estudo sobre engajamento utilizando tarefas adaptadas e apresentadas em softwares lúdicos e convencionais de aprendizado.

Os principais desafios que envolveram o desenvolvimento da pesquisa relatada nesta tese foram os seguintes: 1) determinação de um modelo de dificuldade da tarefa adaptado para cada aluno e em conformidade com o procedimento *Matching-to-Sample*; 2) determinação de um mecanismo de busca que fosse coerente com o procedimento *Matching-to-Sample* seguindo um fluxo de dificuldade; e 3) construção de experimentos de validação tanto no âmbito computacional quanto no âmbito de estudo com participantes do domínio de aplicação.

Originalmente, este trabalho focou-se na alfabetização, mas também pode ser aplicado em outras áreas do conhecimento com algumas adaptações técnicas, como ajustar, adicionar e remover os fatores de dificuldade das tarefas de ensino. Por fim, este estudo pretende auxiliar no ensino de leitura e servir de suporte aos educadores.

6.1. Contribuições científicas

Foram as seguintes as contribuições científicas geradas por esta pesquisa:

- Revista – Qualis B1 ano 2016, em Engenharia IV e A2, ano 2016, em Ciência da Computação
 - Optimizing tasks generation for children in the early stages of literacy teaching: a study using bio-inspired metaheuristics. *Soft Computing*, 1–14, 2018.
- Revista – Qualis B1 ano 2016 em Ensino e Qualis B5 ano 2016 Engenharia IV
 - Um relato de experiência de capacitação para a criação de jogos sérios em cursos de computação com foco em projetos reais. *RENOTE – Revista de Novas Tecnologias na Educação*, 14(1), 2016.
- Revista – Qualis B3 ano 2016 em Educação
 - Jogos digitais e aplicações web como forma de engajamento no ensino da leitura utilizando tarefas adaptativas matching-to-sample. *IE Comunicaciones: revista iberoamericana de informática educativa. Informatica Educativa Comunicaciones*, 27(27), 2018.
- Congresso Nacionais e Internacionais
 - Engagement in digital games and web applications using adaptive matching-to-sample tasks in teaching reading. In *2017 International*

Symposium on Computers in Education (SIEE) (pp. 1–6). IEEE. 2017.
(Qualis B3 de conferência em ciência da computação).

- Um framework para a geração de repertórios de ensino individualizado baseado em dificuldade adaptativa. In *XLVIII SBPO Simpósio Brasileiro de Pesquisa Operacional*. Vitória. 2016. (Qualis B4 de conferência em ciência da computação).
- ABSR: An algorithm based on similarity rules assisted by bio-inspired algorithms for teaching context. In *XLVIII SBPO Simpósio Brasileiro de Pesquisa Operacional*. Vitória. 2016. (Qualis B4 de conferência em ciência da computação).

6.2. Trabalhos futuros

A abordagem para a geração de tarefas adaptadas na presente tese cumpriu todos os objetivos previstos, contudo, os métodos propostos podem evoluir em técnicas e funcionalidades para melhorias significativas. A seguir, alguns trabalhos futuros são apontados:

- Ajuste automático dos parâmetros educacionais. Métodos para ajustar quais palavras devem ser ensinadas, quais os tipos de tarefas devem ser usados em uma determinada situação, ou qual fluxo de dificuldade deve ser inserido no sistema, podem ser valiosos para acelerar a configuração prévia dos parâmetros do conteúdo de ensino.
- Comparações com os métodos pedagógicos tradicionais. Métodos para o ensino da leitura são vários, neste contexto, métodos mais tradicionais, como o ensino em sala de aula, podem ser comparados com o método computacional aplicado nesta tese.
- Novas propostas de sistemas para auxiliar instrutores em ambientes voltados para internet. Para evitar a complexidade do desenvolvimento das atividades de ensino junto com a infra-estrutura necessária, optou-se por desenvolver um sistema de geração de tarefas mono-usuário. Um sistema web amplo e eficiente poderia ser desenvolvido a fim de disponibilizar as características apresentadas nesta tese para domínios maiores e para um maior número participantes.
- Aplicação desta abordagem com outros idiomas ou contextos educacionais semelhantes. A utilização da abordagem desta tese se focou no ensino de palavras do vocabulário português-brasileiro, contudo, o procedimento *Matching-to-Sample* pode ser usado para o ensino de outros idiomas. Existe a possibilidade de ensinar também a matemática,

geografia, música e qualquer disciplina que possui símbolos visuais e efeitos sonoros. Nestes casos, a fórmula de dificuldade e a função de fitness poderiam ser adaptadas para o domínio de aprendizado pretendido.

Essas melhorias irão contribuir para tornar esta abordagem mais abrangente, escalável e mais atrativo para alunos e tutores de diferentes disciplinas de ensino.

Referências Bibliográficas

- Abuiziah, I., & Shakarneh, N. (2014). A Review of Genetic Algorithm Optimization: Operations and Applications to Water Pipeline Systems. *World Academy of Science, Engineering and Technology, International Journal of Mathematical, Computational, Physical, Electrical and Computer Engineering*, 7(12), 1782–1788.
- Al-Shamri, M. Y. H. (2014). Power coefficient as a similarity measure for memory-based collaborative recommender systems. *Expert Systems with Applications*, 41(13), 5680–5688. <https://doi.org/10.1016/J.ESWA.2014.03.025>
- Andrade, J. M., Laros, J. A., & Gouveia, V. V. (2010). O uso da Teoria de Resposta ao Item em avaliações educacionais: diretrizes para pesquisadores. *Avaliação Psicológica*, 9(3), 421–435.
- Baker, F. B., & Al-Karni, A. (1991). A Comparison of Two Procedures for Computing IRT Equating Coefficients. *Journal of Educational Measurement*, 28(2), 147–162. <https://doi.org/10.1111/j.1745-3984.1991.tb00350.x>
- Battaiola, A. L., Martins, F. E., Lívia, L., & Barbosa, R. (2008). O uso da linguagem cinematográfica nos jogos eletrônicos como elemento narrativo. In Carla Coscarelli, Francisco Marinho, Roger Tavares, & Zenilton Patrocínio (Eds.), *VII Brazilian Symposium on Computer Games and Digital Entertainment* (pp. 89–96). Belo Horizonte: Sociedade Brasileira de Computação – SBC.
- Baylari, A., & Montazer, G. A. (2009). Design a personalized e-learning system based on item response theory and artificial neural network approach. *Expert Systems with Applications*, 36(4), 8013–8021. <https://doi.org/10.1016/j.eswa.2008.10.080>
- Bermúdez, A. (2017). 54% dos alunos que deveriam estar alfabetizados têm problemas para ler - Notícias - UOL Educação. Retrieved October 26, 2017, from <https://educacao.uol.com.br/noticias/2017/10/25/54-dos-alunos-que-deveriam-estar-alfabetizados-tem-problemas-para-ler.htm>
- Blanchard, E. G. (2014). Socio-Cultural Imbalances in AIED Research: Investigations, Implications and Opportunities. *International Journal of Artificial Intelligence in Education*, 25(2), 204–228. <https://doi.org/10.1007/s40593-014-0027-7>
- Bolkan, S. (2016). The Importance of Instructor Clarity and Its Effect on Student Learning: Facilitating Elaboration by Reducing Cognitive Load. *Communication Reports*, 29(3), 152–162. <https://doi.org/10.1080/08934215.2015.1067708>

- Bortoloti, R., & Rose, J. C. C. De. (2007). Medida do grau de relacionamento entre estímulos equivalentes. *Psicologia: Reflexão e Crítica*, 20(2), 252–258. <https://doi.org/10.1590/S0102-79722007000200011>
- Butz, M. V, Sastry, K., & Goldberg, D. E. (2003). Tournament Selection: Stable Fitness Pressure in XCS. *Genetic and Evolutionary Computation — GECCO 2003*. Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-45110-2_83
- Cabada, R. Z., Barrón Estrada, M. L., & Reyes García, C. A. (2011). EDUCA: A web 2.0 authoring tool for developing adaptive and intelligent tutoring systems using a Kohonen network. *Expert Systems with Applications*, 38(8), 9522–9529. <https://doi.org/10.1016/j.eswa.2011.01.145>
- Calcada, D., Rosa, A., Duarte, L. C., & Lopes, V. V. (2010). Comparison of GA and PSO performance in parameter estimation of microbial growth models: A case-study using experimental data. *IEEE Congress on Evolutionary Computation*. IEEE. <https://doi.org/10.1109/cec.2010.5586489>
- Cavalcante, S. N. (1997). Abordagem biocomportamental: síntese da análise do comportamento? *Psicologia: Reflexão e Crítica*, 10(2), 263–273. <https://doi.org/10.1590/s0102-79721997000200006>
- Cezarotto, M. A., & Battaiola, A. L. (2016). Game Design Recommendations Focusing on Children with Developmental Dyscalculia. In P. Zaphiris & A. Ioannou (Eds.), *Learning and Collaboration Technologies* (1st ed., pp. 463–473). Toronto: Springer, Cham. https://doi.org/10.1007/978-3-319-39483-1_42
- Chang, T.-Y., & Ke, Y.-R. (2013). A personalized e-course composition based on a genetic algorithm with forcing legality in an adaptive learning system. *Journal of Network and Computer Applications*, 36(1), 533–542. <https://doi.org/10.1016/j.jnca.2012.04.002>
- Chen, C.-M. (2008). Intelligent web-based learning system with personalized learning path guidance. *Computers & Education*, 51(2), 787–814. <https://doi.org/10.1016/j.compedu.2007.08.004>
- Chen, C.-M., & Duh, L.-J. (2008). Personalized web-based tutoring system based on fuzzy item response theory. *Expert Systems with Applications*, 34(4), 2298–2315. <https://doi.org/10.1016/j.eswa.2007.03.010>
- Chen, C. H., & Law, V. (2015). Scaffolding individual and collaborative game-based learning in learning performance and intrinsic motivation. *Computers in Human Behavior*, 55, 1201–1212. <https://doi.org/10.1016/j.chb.2015.03.010>
- Chen, L. X., & Sun, C. T. (2016). Self-regulation influence on game play flow state. *Computers*

- in Human Behavior*, 54, 341–350. <https://doi.org/10.1016/j.chb.2015.08.020>
- Cheng, I., Shen, R., & Basu, A. (2008). An Algorithm for Automatic Difficulty Level Estimation of Multimedia Mathematical Test Items. *2008 Eighth IEEE International Conference on Advanced Learning Technologies*. IEEE. <https://doi.org/10.1109/icalt.2008.105>
- Chin, D. B., Dohmen, I. M., & Schwartz, D. L. (2013). Young Children Can Learn Scientific Reasoning with Teachable Agents. *IEEE Transactions on Learning Technologies*, 6(3), 248–257. <https://doi.org/10.1109/tlt.2013.24>
- Christudas, B. C. L., Kirubakaran, E., & Thangaiah, P. R. J. (2016). An evolutionary approach for personalization of content delivery in e-learning systems based on learner behavior forcing compatibility of learning materials. *Telematics and Informatics*. <https://doi.org/10.1016/j.tele.2017.02.004>
- Chrysafiadi, K., & Virvou, M. (2012). Evaluating the integration of fuzzy logic into the student model of a web-based learning environment. *Expert Systems with Applications*, 39(18), 13127–13134. <https://doi.org/10.1016/j.eswa.2012.05.089>
- Conejo, R., Guzmán, E., Cruz, P. J. L., & Barros, B. (2014). An empirical study on the quantitative notion of task difficulty. *Expert Systems with Applications*, 41(2), 594–606. <https://doi.org/10.1016/j.eswa.2013.07.084>
- De-Marcos, L., Domínguez, A., Saenz-De-Navarrete, J., & Pagés, C. (2014). An empirical study comparing gamification and social networking on e-learning. *Computers and Education*, 75, 82–91. <https://doi.org/10.1016/j.compedu.2014.01.012>
- de Souza, D. G., de Rose, J. C., Faleiros, T. C., Bortoloti, R., Hanna, E. S., & 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 Psicologica = International Journal of Psychology and Psychological Therapy*, 9(1), 19–44. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/19960112>
- de Souza, S., Goyos, C., Silvaes, E. F., & Saunders, R. R. (2007). Emergence of Printing and Spelling Skills from Constructed-Response Matching-to-Sample Instruction (CRMTS). *European Journal of Behavior Analysis*, 8(1), 49. Retrieved from http://www.ejoba.org/PDF/2007_1/deSouza_Goyos_Silvaes_Saunders_2007.pdf%5Cnh
<http://www.ejoba.org/cgi-bin/search.cgi?volume=8&issue=1>
- Debert, P., & Andery, M. A. P. A. (2016). Discriminação condicional: definições, procedimentos e dados recentes. *Revista Brasileira de Análise Do Comportamento*, 2(1). <https://doi.org/10.18542/rebac.v2i1.801>

- Dixon, M. R., Belisle, J., Stanley, C. R., Daar, J. H., & Williams, L. A. (2016). Derived Equivalence Relations of Geometry Skills in Students with Autism: an Application of the PEAK-E Curriculum. *The Analysis of Verbal Behavior*, 32(1), 38–45. <https://doi.org/10.1007/s40616-016-0051-9>
- Dorigo, M., & Blum, C. (2005). Ant colony optimization theory: A survey. *Theoretical Computer Science*, 344(2–3), 243–278. <https://doi.org/10.1016/j.tcs.2005.05.020>
- Dorigo, M., Maniezzo, V., & Colorni, A. (1996). The Ant System: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics—Part B IEEE Transactions on Systems, Man, and Cybernetics, Part–B*, 2626(11), 1–131. Retrieved from <http://iridia.ulb.ac.be/dorigo/dorigo.html>
- Dung, P. Q., & Florea, A. M. (2012). A literature-based method to automatically detect learning styles in learning management systems. *Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics - WIMS '12*, 46, 1. <https://doi.org/10.1145/2254129.2254186>
- Ekin, C. C., Cagiltay, K., & Karasu, N. (2018). Effectiveness of smart toy applications in teaching children with intellectual disability. *Journal of Systems Architecture*, 89, 41–48. <https://doi.org/10.1016/J.SYSARC.2018.07.001>
- Fazal Rehman Shamil. (2018). Min Max Normalization of data in data mining. Retrieved June 5, 2018, from <https://t4tutorials.com/min-max-normalization-of-data-in-data-mining/>
- Feldman, J., Monteserin, A., & Amandi, A. (2014). Automatic detection of learning styles: state of the art. *Artificial Intelligence Review*, 44(2), 157–186. <https://doi.org/10.1007/s10462-014-9422-6>
- Galvão, O. de F., & Barros, R. da S. (2001). Curso de Introdução a Análise Experimental do Comportamento. Retrieved from <http://pt.scribd.com/doc/3235127/Curso-de-Introducao-a-Analise-Experimental-do-Comportamento>
- Gerard, C. J., Mackay, H. A., Thompson, B., & McIlvane, W. J. (2014). Rapid generation of balanced trial distributions for discrimination learning procedures: A technical note. *Journal of the Experimental Analysis of Behavior*, 101(1), 171–178. <https://doi.org/10.1002/jeab.58>
- Goyos, C. (2012). Equivalence Class Formation Via Common Reinforcers Among Preschool Children. *The Psychological Record*, 50(4). Retrieved from <http://opensiuc.lib.siu.edu/tpr/vol50/iss4/2>
- Hamari, J., & Koivisto, J. (2014). Measuring flow in gamification: Dispositional Flow Scale-2. *Computers in Human Behavior*, 40, 133–143. <https://doi.org/10.1016/j.chb.2014.07.048>

- Hanna, E. S., Karino, C. A., Araújo, V. T., & Souza, D. D. G. De. (2010). Leitura recombinaiva de pseudopalavras impressas em pseudoalfabeto: similaridade entre palavras e extensão da unidade ensinada. *Psicologia USP*, 21(400359), 275–311. <https://doi.org/10.1590/S0103-65642010000200005>
- Hanus, M. D., & Fox, J. (2014). Assessing the effects of gamification in the classroom: A longitudinal study on intrinsic motivation, social comparison, satisfaction, effort, and academic performance. *Computers & Education*, 80, 152–161. <https://doi.org/10.1016/j.compedu.2014.08.019>
- Haworth, R., & Sedig, K. (2011). The importance of design for educational games. In A. Mendez-Vilas (Ed.), *Education in a technological world: communicating current and emerging research and technological efforts* (pp. 518–522). Formatex Research Center. Retrieved from <http://www.formatex.info/ict/book/518-522.pdf>
- Holland, J. H. (1975). *Adaptation in natural and artificial system. The University of Michigan Press.*
- Holmes, M., Latham, A., Crockett, K., & OShea, J. D. (2018). Near Real-Time Comprehension Classification with Artificial Neural Networks: Decoding e-Learner Non-Verbal Behavior. *IEEE Transactions on Learning Technologies*, 11(1), 5–12. <https://doi.org/10.1109/TLT.2017.2754497>
- Hong, H.-Y., & Chai, C. S. (2017). Principle-based design: Development of adaptive mathematics teaching practices and beliefs in a knowledge building environment. *Computers & Education*, 115, 38–55. <https://doi.org/10.1016/J.COMPEDU.2017.07.011>
- Hong, J.-C., Hwang, M.-Y., Chen, W.-C., Lee, C.-C., Lin, P.-H., & Chen, Y.-L. (2013). Comparing the retention and flow experience in playing Solitary and Heart Attack games of San Zi Jing: A perspective of Dual Process Theory. *Computers & Education*, 69, 369–376. <https://doi.org/10.1016/j.compedu.2013.07.027>
- Huang, M., Huang, H., & Chen, M. (2007). Constructing a personalized e-learning system based on genetic algorithm and case-based reasoning approach. *Expert Systems with Applications*, 33(3), 551–564. <https://doi.org/10.1016/j.eswa.2006.05.019>
- Hwang, G.-J., & Wang, S.-Y. (2016). Single loop or double loop learning: English vocabulary learning performance and behavior of students in situated computer games with different guiding strategies. *Computers & Education*, 102, 188–201. <https://doi.org/10.1016/j.compedu.2016.07.005>
- IBGE. (2011). Séries Estatísticas & Séries Históricas. Retrieved December 30, 2016, from <http://seriesestatisticas.ibge.gov.br/series.aspx?no=4&op=0&vcodigo=PD321&t=pessoa>

s-5-anos-mais-idade-nao

- Irvine, A., Drew, P., & Sainsbury, R. (2013). “Am I not answering your questions properly?” Clarification, adequacy and responsiveness in semi-structured telephone and face-to-face interviews. *Qualitative Research*, 13(1), 87–106. <https://doi.org/10.1177/1468794112439086>
- Izakian, H., Ladani, B. T., Abraham, A., & Snášel, V. (2010). A discrete particle swarm optimization approach for grid job scheduling. *International Journal of Innovative Computing*, 6(9), 1–9. Retrieved from <http://www.mirlabs.org>
- Jain, P., Kumaraguru, P., & Joshi, A. (2016). Other times, other values: leveraging attribute history to link user profiles across online social networks. *Social Network Analysis and Mining*, 6(1), 85. <https://doi.org/10.1007/s13278-016-0391-4>
- Jegatha Deborah, L., Baskaran, R., & Kannan, A. (2012). Learning styles assessment and theoretical origin in an E-learning scenario: a survey. *Artificial Intelligence Review*, 1–19. <https://doi.org/10.1007/s10462-012-9344-0>
- Jiao, W., Liu, G., & Liu, D. (2008). Elite Particle Swarm Optimization with mutation. *2008 Asia Simulation Conference - 7th International Conference on System Simulation and Scientific Computing*. IEEE. <https://doi.org/10.1109/asc-icsc.2008.4675471>
- Juliano, M. C. (2009). Análise dos efeitos de uma programação de ensino sobre o desempenho de estudantes em um curso superior de administração de empresas. *Revista de Educação*, 13, 23–28.
- Kar, A. K. (2016). Bio inspired computing – A review of algorithms and scope of applications. *Expert Systems with Applications*, 59, 20–32. <https://doi.org/10.1016/J.ESWA.2016.04.018>
- Kennedy J, E. R. (1995). Particle swarm optimization. In *IEEE International Conference on Neural Networks* (p. 1942–8.).
- Kim, K., Schmierbach, M. G., Bellur, S., Chung, M.-Y., Fraustino, J. D., Dardis, F., & Ahern, L. (2015). Is it a sense of autonomy, control, or attachment? Exploring the effects of in-game customization on game enjoyment. *Computers in Human Behavior*, 48, 695–705. <https://doi.org/10.1016/j.chb.2015.02.011>
- Klinkenberg, S., Straatemeier, M., & van der Maas, H. L. J. L. J. (2011). Computer adaptive practice of Maths ability using a new item response model for on the fly ability and difficulty estimation. *Computers & Education*, 57(2), 1813–1824. <https://doi.org/10.1016/j.compedu.2011.02.003>
- Kostolányová, K., & Šarmanová, J. (2014). Use of Adaptive Study Material in Education in E-

- learning Environment. *Electronic Journal of E-Learning*, 12(2), 172–182.
- Krynicky, K., Jaen, J., & Navarro, E. (2016). An ACO-based personalized learning technique in support of people with acquired brain injury. *Applied Soft Computing*, 47, 316–331. <https://doi.org/10.1016/j.asoc.2016.04.039>
- Kumar, R., & Kim, J. (2014). Special Issue on Intelligent Support for Learning in Groups. *International Journal of Artificial Intelligence in Education*, 24(1), 1–7. <https://doi.org/10.1007/s40593-013-0013-5>
- Li, J.-W., Chang, Y.-C., Chu, C.-P., & Tsai, C.-C. (2012). A self-adjusting e-course generation process for personalized learning. *Expert Systems with Applications*, 39(3), 3223–3232. <https://doi.org/10.1016/j.eswa.2011.09.009>
- Li, M. L. M., Ogata, H., Hou, B. H. Bin, Hashimoto, S., Uosaki, N., Liu, Y. L. Y., & Yano, Y. (2010). Development of Adaptive Vocabulary Learning via Mobile Phone E-mail. *2010 6th IEEE International Conference on Wireless, Mobile, and Ubiquitous Technologies in Education*, 34–41. <https://doi.org/10.1109/WMUTE.2010.9>
- Lin, C. F., Yeh, Y. C., Hung, Y. H., & Chang, R. I. (2013). Data mining for providing a personalized learning path in creativity: An application of decision trees. *Computers and Education*, 68, 199–210. <https://doi.org/10.1016/j.compedu.2013.05.009>
- Liu, Y. C., & Chen, P. J. (2012). Discovering discriminative test items for achievement tests. *Expert Systems with Applications*, 39(1), 1426–1434. <https://doi.org/10.1016/j.eswa.2011.08.026>
- Machado, C. (2016). Inteligência Artificial – Otimização por enxame de partículas (PSO). Retrieved August 16, 2018, from <https://cleitonmachado.wordpress.com/2016/08/07/inteligencia-artificial-otimizacao-por-enxame-de-particulas-pso/>
- Mackay, H. A. (1985). Stimulus equivalence in rudimentary reading and spelling. *Analysis and Intervention in Developmental Disabilities*, 5(4), 373–387. [https://doi.org/10.1016/0270-4684\(85\)90006-0](https://doi.org/10.1016/0270-4684(85)90006-0)
- Marković, S., & Jovanović, N. (2011). Learning style as a factor which affects the quality of e-learning. *Artificial Intelligence Review*, 38(4), 303–312. <https://doi.org/10.1007/s10462-011-9253-7>
- Marques, L. B., de Souza, D. G., Axe, J. B., Sainato, D. M., Barendregt, W., Bekker, T. M., ... Kodak, T. (2013). Behavioral Evaluation of Preference for Game-Based Teaching Procedures. *International Journal of Game-Based Learning*, 3(1), 51–62. <https://doi.org/10.4018/ijgbl.2013010104>

- McGonigal, J. (2011). *Reality Is Broken: Why Games Make Us Better and How They Can Change the World*. (1 edition). Penguin Press.
- Méndez, J. A., & González, E. J. (2013). A control system proposal for engineering education. *Computers & Education*, *68*, 266–274. <https://doi.org/10.1016/j.compedu.2013.05.014>
- Millán, E., Descalço, L., Castillo, G., Oliveira, P., & Diogo, S. (2013). Using Bayesian networks to improve knowledge assessment. *Computers & Education*, *60*(1), 436–447. <https://doi.org/10.1016/j.compedu.2012.06.012>
- Mitchell, M., & Taylor, C. E. (1999). Evolutionary Computation: An Overview. *Annual Review of Ecology and Systematics*, *30*(1), 593–616. <https://doi.org/10.1146/annurev.ecolsys.30.1.593>
- Mizukami, M. d. (1986). *ENSINO: As abordagens do processo*. São Paulo: EPU.
- Mueller, M. M., Olmi, D. J., & Saunders, K. J. (2000). Recombinative generalization of within-syllable units in prereading children. *Journal of Applied Behavior Analysis*, *33*(4), 515–531. <https://doi.org/10.1901/jaba.2000.33-515>
- Nerino, G., Monteiro, D. C., Pereira, A. B. C. P. B. C., Barros, E. S., & Marques, L. B. (2012). Máquina de aprendizagem como ferramenta de auxílio na análise comportamental no ensino da leitura. *Renote*, *10*(1), 1–11. Retrieved from <http://seer.ufrgs.br/index.php/renote/article/view/30853>
- Nerino G Jr, Fontes M, Monteiro D, & Santana A. (2016). ABSR: An algorithm based on similarity rules assisted by bio-inspired algorithms for teaching context. In *XLVIII SBPO Simpósio Brasileiro de Pesquisa Operacional*. Vitória.
- Nešić, S., Gašević, D., Jazayeri, M., & Landoni, M. (2011). A Learning Content Authoring Approach based on Semantic Technologies and Social Networking: an Empirical Study. *Educational Technology & Society*, *14*(4), 35–48. Retrieved from http://www.ifets.info/journals/14_4/5.pdf
- Niwattanakul, S., Singthongchai, J., Naenudorn, E., & Wanapu, S. (2013). Using of Jaccard Coefficient for Keywords Similarity. Retrieved from <https://www.semanticscholar.org/paper/Using-of-Jaccard-Coefficient-for-Keywords-Niwattanakul-Singthongchai/db467107ad133e44085780da5296b840e2a32e9e>
- Nogueira, R. C. C., & Both, I. J. (2012). A importância do tutor em Educação a Distância (EaD). *Caderno Intersaberes*, *1*, 92–102.
- Orlando, A. F. (2009). *Uma infra-estrutura computacional para o gerenciamento de programas de ensino individualizados*. Universidade Federal de São Carlos.
- Özyurt, H., Özyurt, Ö., Baki, A., & Güven, B. (2012). Integrating computerized adaptive testing

- into UZWEBMAT: Implementation of individualized assessment module in an e-learning system. *Expert Systems with Applications*, 39(10), 9837–9847. <https://doi.org/10.1016/j.eswa.2012.02.168>
- Panchal, G., & Panchal, D. (2015). *Solving NP hard Problems using Genetic Algorithm*. Retrieved from www.ijcsit.com
- Pedri, V. (2012). A Importância do Tutor no Processo de Aprendizagem na Modalidade de EAD. Retrieved from <http://www.portaleducacao.com.br/educacao/artigos/25184/a-importancia-do-tutor-no-processo-de-aprendizagem-na-modalidade-de-ead>
- Pereira, A. B. C., de Souza, G. N., Monteiro, D. C., & Marques, L. B. (2013). A Fuzzy System for Educational Tasks for Children with Reading and Writing Disabilities (pp. 57–78). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-36651-2_4
- Pereira A, Nerino G Jr, Monteiro C, Barros E, Costa H, Nascimento P, ... Bessa R. (2012). A AIED Game to help children with learning disabilities in literacy in the Portuguese language. In *Simpósio Brasileiro de Jogos e Entretenimento Digital - SBGames* (pp. 134–143). Brasília.
- Prensky, M. (2003). Digital game-based learning. *Computers in Entertainment*, 1(1), 21. <https://doi.org/10.1145/950566.950596>
- Puzenat, D., & Verlut, I. (2010). Behavior Analysis through Games Using Artificial Neural Networks. *2010 Third International Conference on Advances in Computer-Human Interactions*. IEEE. <https://doi.org/10.1109/achi.2010.26>
- Reis, T. de S., Souza, D. das G. de, & Rose, J. C. de. (2009). Avaliação de um programa para o ensino de leitura e escrita. *Estudos Em Avaliação Educacional*, 20(44), 425. <https://doi.org/10.18222/ae204420092038>
- Ren, Z., Liu, M., Ye, C., Wang, X., & Wang, C. (2010). Realization and research of paperless examination system based on WEB. In *Proceedings - 2nd IEEE International Conference on Advanced Computer Control, ICACC 2010* (Vol. 4, pp. 90–93).
- Ricarte, T. A. M. (2013). *Teste adaptativo computadorizado nas avaliações educacionais e psicológicas*. Universidade de Sao Paulo Sistema Integrado de Bibliotecas - SIBiUSP. <https://doi.org/10.11606/d.55.2013.tde-16052013-143315>
- Richardson, U., & Lyytinen, H. (2014). The GraphoGame Method: The Theoretical and Methodological Background of the Technology-Enhanced Learning Environment for Learning to Read. *Human Technology: An Interdisciplinary Journal on Humans in ICT Environments*, 10(1), 39–60. <https://doi.org/10.17011/ht/urn.201405281859>
- Rodrigo. (2009). Conceitos básicos da AC – Parte 5 – O reforçamento.

- Ronimus, M., Kujala, J., Tolvanen, A., & Lyytinen, H. (2014). Children's engagement during digital game-based learning of reading: The effects of time, rewards, and challenge. *Computers & Education, 71*, 237–246. <https://doi.org/10.1016/j.compedu.2013.10.008>
- Ronimus, M., & Richardson, U. (2014). Digital game-based training of early reading skills: overview of the GraphoGame method in a highly transparent orthography / Entrenamiento de habilidades de lectura tempranas basado en un juego digital: visión general del método GraphoGame en una ortograf. *Estudios de Psicología, 35*(3), 648–661. <https://doi.org/10.1080/02109395.2014.974424>
- Rose, J. C. de, & Bortoloti, R. (2007). A equivalência de estímulos como modelo do significado. *Acta Comportamentalia, 15*(SPE), 83–102.
- Rose, J., Souza, D. G. de, Rossito, A. L., & Rose, T. M. S. de. (2012). Aquisição de leitura após história de fracasso escolar: equivalência de estímulos e generalização. *Psicologia: Teoria e Pesquisa, 5*(3), 325–346.
- Sampayo-Vargas, S., Cope, C., He, Z., & Byrne, G. (2013). The effectiveness of adaptive difficult adjustment in an educational computer game. *Computers & Education, In Press*. <https://doi.org/10.1016/j.compedu.2013.07.004>
- Seiter, L., & Foreman, B. (2013). Modeling the learning progressions of computational thinking of primary grade students. In *Proceedings of the ninth annual international ACM conference on International computing education research - ICER '13* (p. 59). New York, New York, USA: ACM Press. <https://doi.org/10.1145/2493394.2493403>
- Sharkness, J. (2014). Item Response Theory: Overview, Applications, and Promise for Institutional Research. *New Directions for Institutional Research, 2014*(161), 41–58. <https://doi.org/10.1002/ir.20066>
- Shen, V. R. L., Yang, C.-Y., Wang, Y.-Y., & Lin, Y.-H. (2012). Application of high-level fuzzy Petri nets to educational grading system. *Expert Systems with Applications, 39*(17), 12935–12946. <https://doi.org/10.1016/j.eswa.2012.05.030>
- Sidman, M., & Tailby, W. (2006). Discriminação Condicional VS Emparelhamento com o Modelo: Uma Expansão do Paradigma de Teste. *Revista Brasileira de Análise Do Comportamento, 2*(1), 115–139.
- Siqueira, E. S., Barros, E. S., Monteiro, D. C., Souza, D. das G. de, & Marques, L. B. (2012). *ALE RPG - Jogo Digital para Aprendizagem de Crianças em Leitura e Escrita. RENOTE* (Vol. 10). CINTED/UFRGS. Retrieved from <http://seer.ufrgs.br/index.php/renote/article/view/30789/19194>
- Skinner, B. F. (1950). Are theories of learning necessary? *Psychological Review, 57*(4), 193–

216. <https://doi.org/10.1037/H0054367>
- Smith, A. E., & Coit, D. W. (1996). Penalty Functions. In T. Baeck, D. Fogel, & Z. Michalewicz (Eds.), *Handbook of Evolutionary Computation*. Pennsylvania: University of Pittsburgh.
- Smith, G. G., Li, M., Drobisz, J., Park, H.-R., Kim, D., & Smith, S. D. (2013). Play games or study? Computer games in eBooks to learn English vocabulary. *Computers & Education*, *69*, 274–286. <https://doi.org/10.1016/j.compedu.2013.07.015>
- Smith, T. (2001). Discrete Trial Training in the Treatment of Autism. *Focus on Autism and Other Developmental Disabilities*, *16*(2), 86–92. <https://doi.org/10.1177/108835760101600204>
- Strasser, S., Goodman, R., Sheppard, J., & Butcher, S. (2016). A New Discrete Particle Swarm Optimization Algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference* (pp. 53–60). Denver. <https://doi.org/10.1145/2908812.2908935>
- Sun, K.-T., Chen, Y.-J., Tsai, S.-Y., & Cheng, C.-F. (2008). Creating IRT-Based Parallel Test Forms Using the Genetic Algorithm Method. *Applied Measurement in Education*, *21*(2), 141–161. <https://doi.org/10.1080/08957340801926151>
- The Huffington Post. (2014). Report: 250 million school age kids can't read. Retrieved December 30, 2016, from <https://www.sott.net/article/272930-Report-250-million-school-age-kids-cant-read>
- Todorov, J. C. (1982). *Behaviorismo e Análise Experimental do Comportamento*. Brasília: Universidade de Brasília. Retrieved from <http://professor.ucg.br/SiteDocente/admin/arquivosUpload/15372/material/Behaviorismo e a Anál Exp Cpto.pdf>
- Understanding the Flow Channel in Game Design. (2015). Retrieved August 19, 2018, from <http://fireside.gamejolt.com/post/understanding-the-flow-channel-in-game-design-ghuttxg6>
- UNESCO. (2012). *Technical note prepared for the Education for All Global Monitoring Report 2012*. Retrieved from <http://unesdoc.unesco.org/images/0021/002193/219349E.pdf>
- UNESCO. (2014). *Around 250 million children of primary school-age are not reaching a minimum learning standard*. Retrieved from <http://unesdoc.unesco.org/images/0022/002287/228723E.pdf>
- Verdú, E., Verdú, M. J., Regueras, L. M., De Castro, J. P., & García, R. (2012). A genetic fuzzy expert system for automatic question classification in a competitive learning environment. *Expert Systems with Applications*, *39*(8), 7471–7478. <https://doi.org/10.1016/j.eswa.2012.01.115>

- Viswanathan, S. A., & VanLehn, K. (2018). Using the Tablet Gestures and Speech of Pairs of Students to Classify Their Collaboration. *IEEE Transactions on Learning Technologies*, *11*(2), 230–242. <https://doi.org/10.1109/TLT.2017.2704099>
- Walker, B. D., & Rehfeldt, R. A. (2012). An evaluation of the stimulus equivalence paradigm to teach single-subject design to distance education students via blackboard. *Journal of Applied Behavior Analysis*, *45*(2), 329–344. <https://doi.org/10.1901/jaba.2012.45-329>
- Wang, J., Jou, M., Lv, Y., & Huang, C.-C. (2018). An investigation on teaching performances of model-based flipping classroom for physics supported by modern teaching technologies. *Computers in Human Behavior*, *84*, 36–48. <https://doi.org/10.1016/J.CHB.2018.02.018>
- Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image Quality Assessment: From Error Visibility to Structural Similarity. In *IEEE Transactions on Image Processing* (Vol. 13, pp. 600–612). <https://doi.org/10.1109/TIP.2003.819861>
- Wauters, K., Desmet, P., & Van Den Noortgate, W. (2012). Item difficulty estimation: An auspicious collaboration between data and judgment. *Computers and Education*, *58*, 1183–1193. <https://doi.org/10.1016/j.compedu.2011.11.020>
- William M K Trochim. (2006). Social Research Methods - Knowledge Base - Likert Scaling. Retrieved August 12, 2018, from <http://www.socialresearchmethods.net/kb/scallik.php>
- Yarkoni, T., Balota, D., & Yap, M. (2008). Moving beyond Coltheart's N: A new measure of orthographic similarity. *Psychonomic Bulletin & Review*, *15*(5), 971–979. <https://doi.org/10.3758/PBR.15.5.971>

Apêndices

Esta seção tem como objetivo mostrar documentos e informações complementares desenvolvidos nesta pesquisa.

Apêndice I. Algoritmo de ordenação de tarefas baseado no Fluxo de Dificuldade.

Este apêndice exhibe o Algoritmo 1 em pseudocódigo, que realiza a ordenação das tarefas com base no o Fluxo de Dificuldade inserido após execução dos algoritmos bio-inspirados. Os algoritmos e técnicas de inteligência computacional foram desenvolvidos através da linguagem de programação MatLab ver2016.

Algoritmo 1: Função <i>OrdenarTarefasBaseadoNoFluxoDeDificuldade(Tarefas,DificuldadeDasTarefas,FluxoDeDificuldade)</i>	
1:	TarefasOrdenadas = Tarefas
2:	Taux = Tarefas
3:	D = DificuldadeDasTarefas
4:	DificuldadeDasTarefasOrdenadas = DificuldadeDasTarefas
5:	[Ds, Di] = ordenar(D) //retorna um Di com índices ordenados que especifica quais dos elementos //de D foram rearranjados na obtenção do vetor Ds . Ds são os valores de dificuldade ordenados.
6:	F = FluxoDeDificuldade
7:	[Fs, Fi] = ordenar(F) //retorna um Fi com índices ordenados que especifica quais dos elementos //de F foram rearranjados na obtenção do vetor Fs . Fs são os valores do fluxo de dificuldade ordenados.
8:	Para cada <i>i</i> Elemento do FluxoDeDificuldade ,
9:	{
10:	Taux { i } = Tarefas { Di [i] }
11:	DificuldadeDasTarefasOrdenadas { Fi [i] } = Ds [i]
12:	TarefasOrdenadas { Fi [i] } = Taux { i }
13:	}
14:	Retornar TarefasOrdenadas , DificuldadeDasTarefasOrdenadas

Obs: A função `ordenar()`, encontrada na linha 5 e 7, possui uma função similar em <https://www.mathworks.com/help/matlab/ref/sort.html>.

Apêndice II. Algoritmo de similaridade entre escolhas textuais.

No contexto educacional, medir a similaridade entre palavras é um componente importante para analisar a percepção de objetos e mensurar a dificuldade de tarefas. Contudo, raramente utiliza-se regras específicas para refinamento do valor de similaridade.

Apresenta-se o Algoritmo de Similaridade Baseado em Regras (ASBR), capaz de gerar a similaridade entre palavras dentro de um contexto de ensino da análise comportamental para o procedimento *Matching-to-sample* (MTS). A similaridade pode ser calculada através de cada regra que possui um peso associado conforme o Algoritmo 2 abaixo,

Algoritmo 2 : ASBR
<p>Entrada : Palavra1, Palavra2, Regras, W, n Saída: S Início Para $i \leftarrow 1$, até n, faça, Se $\text{Regras}[i].(\text{Palavra1}, \text{Palavra2}) = \text{Verdadeiro}$, então $V_{r_i} \leftarrow \text{Regras}[i].\text{CalcularValorDaRegra}(\text{Palavra1}, \text{Palavra2}, W_i)$ FimSe FimPara $S \leftarrow \sum_{i=1}^n W_i V_{r_i}$ Fim</p>

onde: S é a similaridade; W_i é o peso da i -ésima regra; V_{r_i} é o valor da i -ésima regra; n número total de regras.

As seguintes regras foram definidas:

1. Qual a distância de *Levensthein*?
2. Qual a diferença entre o tamanho das palavras?
3. Primeira letra da primeira palavra é igual a primeira letra da segunda palavra?
4. Primeira sílaba da primeira palavra é igual a primeira sílaba da segunda palavra?
5. Última sílaba da primeira palavra é igual a última sílaba da segunda palavra?;
6. Última letra da primeira palavra é igual a última letra da segunda palavra?

Foi realizado um estudo de caso onde as opiniões de similaridade de 9 especialistas foram coletadas através de questionários. Foram utilizadas 25 palavras do programa de ensino chamado ALEPP (Aprendendo a Ler e Escrever em Pequenos Passos) com a coleta de 1400 valores de similaridades entre palavras. Após a coleta de dados foi tirado a média e o erro padrão das opiniões dos psicólogos. A Tabela APII.15 mostra uma amostra de média dos dados sugeridos pelos especialistas e normalizadas em escala probabilística.

Tabela APII.15 – Média da indicação da opinião dos especialistas entre palavras textuais para 15 palavras. Quanto maior o valor, maior a similaridade.

	bolo	tatu	vaca	bico	mala	tubo	pipa	cavalo	apito	luva	tomate	vovô	muleta	fita	pato
bolo	1	0,15	0,15	0,525	0,25	0,525	0,15	0,5	0,2	0,175	0,225	0,3	0,225	0,15	0,25
tatu	0,15	1	0,225	0,15	0,225	0,525	0,2	0,225	0,2	0,175	0,35	0,15	0,5	0,625	0,4
vaca	0,15	0,225	1	0,225	0,45	0,15	0,225	0,6	0,15	0,525	0,2	0,3	0,2	0,25	0,225
bico	0,525	0,15	0,225	1	0,15	0,325	0,275	0,225	0,325	0,15	0,15	0,2	0,15	0,225	0,25
mala	0,225	0,225	0,5	0,15	1	0,15	0,25	0,325	0,15	0,3	0,35	0,15	0,4	0,225	0,25
tubo	0,525	0,525	0,15	0,325	0,15	1	0,15	0,2	0,275	0,225	0,325	0,2	0,275	0,2	0,275
pipa	0,15	0,2	0,225	0,275	0,25	0,15	1	0,2	0,525	0,225	0,15	0,15	0,2	0,35	0,525
cavalo	0,5	0,225	0,525	0,225	0,35	0,2	0,2	1	0,375	0,475	0,25	0,225	0,175	0,15	0,3
apito	0,2	0,2	0,15	0,325	0,15	0,275	0,525	0,375	1	0,15	0,475	0,15	0,275	0,25	0,575
luva	0,15	0,175	0,525	0,15	0,3	0,25	0,225	0,475	0,15	1	0,2	0,2	0,375	0,275	0,15
tomate	0,225	0,325	0,2	0,15	0,35	0,3	0,15	0,25	0,475	0,2	1	0,225	0,35	0,275	0,5
vovô	0,275	0,15	0,275	0,2	0,15	0,2	0,15	0,225	0,15	0,2	0,225	1	0,15	0,15	0,2
muleta	0,225	0,475	0,2	0,15	0,4	0,275	0,2	0,175	0,275	0,375	0,325	0,15	1	0,525	0,25
fita	0,15	0,575	0,25	0,225	0,225	0,2	0,4	0,15	0,25	0,275	0,275	0,15	0,525	1	0,375
pato	0,25	0,4	0,225	0,25	0,225	0,275	0,5	0,3	0,625	0,15	0,5	0,2	0,275	0,55	1

Para realizar a busca dos pesos das regras foi utilizado algoritmos computacionais onde o objetivo foi encontrar um conjunto de 6 pesos que gerem similaridades dentro da margem de erro e próximo da média das similaridades fornecidas pelas opiniões dos especialistas. Desta forma foi encontrado os seguintes pesos para as regras citadas: 0.49, 0.174, 0.064, 0.0772, 0.1702, 0.0246 respectivamente.

As opiniões dos especialistas foram usadas para calcular o RMSE (*Root Mean Square Error*) em relação a métricas de similaridade já utilizadas na literatura. A similaridade de Jaro compara caracteres comuns e é capaz de mensurar a similaridade de perfis de usuários ao longo do tempo com o intuito de destacar as preferências do usuário (Jain, Kumaraguru, & Joshi, 2016). A similaridade Dice compara segmentos de uma palavra considerados iguais e é muito utilizado em sistemas de recomendação de produtos para consumidores (Al-Shamri, 2014). A similaridade de Jaccard é utilizada para calcular a similaridade entre duas *strings* através do tamanho da interseção entre as *strings* e já foi utilizada no campo da correção ortográfica para encontrar erros presentes em textos (Niwattanukul, Singthongchai, Naenudorn, & Wanapu, 2013). A similaridade de Levenshtein calcula a similaridade entre duas *strings* estimando a sequência mínima de operações que transformam uma *string* em outra e é muito utilizada no campo de análise de erros ortográficos (Yarkoni, Balota, & Yap, 2008).

Após as execuções dessas métricas, os seguintes resultados foram gerados: ASBR (0,083), *Levenshtein* (0.096), *Dice* (0.207), *Jaccard* (0.224) e *Jaro* (0.303). Levando em consideração que quanto menor o valor de RMSE, maior proximidade entre a métrica de similaridade da opinião dos especialistas. Logo, o ASBR aproximou-se mais das opiniões de especialistas tornando-o um algoritmo viável para a identificação de similaridade entre escolhas das tarefas de ensino do procedimento MTS.

Apêndice III. Termo de Consentimento Livre e Esclarecido.

TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

Dados de identificação

Título do Projeto: Jogos computacionais para o processo de ensino-aprendizagem de crianças em fase de alfabetização

Pesquisadores Responsáveis: Msc. Gilberto Nerino de Souza Junior; Prof. Dr. Dionne Cavalcante Monteiro, Prof. Dr. Ádamo Lima de Santana.

Instituição dos Pesquisadores Responsáveis: Universidade Federal do Pará - Instituto de Tecnologia.

Telefones para contato: (91) 8255-5350 – Gilberto Nerino de Souza Jr (gilbertonerinojr@gmail.com)

Nome _____ do
estudante: _____

Idade: _____ anos R.G. _____

Responsável legal (quando for o caso): _____

R.G. Responsável legal: _____

O (a) estudante _____ está sendo convidado(a) a participar do projeto de pesquisa **Jogos computacionais para o processo de ensino-aprendizagem de crianças em fase de alfabetização**, de responsabilidade do pesquisador Gilberto Nerino de Souza Junior, sob orientação do Prof. Dr. Dionne Cavalcante Monteiro e Prof. Dr. Ádamo Lima de Santana.

O presente trabalho tem por objetivo medir o engajamento de estudantes em aplicações que buscam ensinar a ler e soletrar utilizando tarefas de ensino. Uma das aplicações é um jogo que busca motivar e tornar o ensino lúdico. Para medir o engajamento do aluno no jogo e na aplicação, é necessário realizar um experimento no laboratório de informática da escola com alunos em fase de alfabetização. A sessão do experimento terá em torno de 1 hora. Dependendo do desempenho da criança durante a aplicação do jogo, haverá a possibilidade de estender a sessão por mais 30 minutos.

O método do experimento será dividido em três momentos: aplicação de questionário prévio, execução da aplicação computacional e do jogo nos grupos de controle e questionário final de avaliação. O experimento ocorrerá no laboratório de informática da escola. Todas as informações obtidas pelo registro ou experimento serão sigilosas.

A realização deste trabalho não acarretará prejuízos quer do ponto de vista acadêmico, quer de qualquer natureza ao estudante participante. O experimento tem como grande meta auxiliar para que o participante adquira competências de leitura e soletração em palavras específicas em um ambiente considerado lúdico, que lhe serão úteis em todo seu percurso acadêmico.

A participação do estudante é **voluntária**, sendo que este consentimento poderá ser retirado a qualquer momento, sem prejuízos a nenhuma das partes envolvidas.

Reitera-se que qualquer informação gerada será sigilosa, sendo de uso exclusivamente científico. A privacidade dos estudantes que puderem participar desta pesquisa será sempre preservada e será assegurado, portanto, o cuidado para que de forma alguma se possa identificar o participante.

Informamos que o(a) senhor(a) não pagará nem será remunerado por sua participação. Garantimos, no entanto, que todas as despesas decorrentes da pesquisa serão ressarcidas, quando devidas e decorrentes especificamente de sua participação na pesquisa.

Qualquer dúvida eventual sobre esta pesquisa poderá ser sanada com um telefonema ou e-mail aos pesquisadores responsáveis (informações disponibilizadas no início deste termo), ou pelo ou Comitê de Ética em Pesquisa do Instituto de Ciências da Saúde da Universidade Federal do Pará (CEP-ICS/UFPA) Complexo de Sala de Aula/ICS, Sala 13 - Campus Universitário, no. 1, Guamá, CEP: 66.075-110. Belém, Pará. Tel/Fax: 3201-7735, E-mail: cepccs@ufpa.br.

Agradecemos sua apreciação deste documento e sua eventual concordância com a participação nesta pesquisa.

Eu, _____, RG nº _____,
responsável legal por _____, RG nº _____
declaro ter sido informado e concordo com a sua participação, como voluntário, no projeto de pesquisa
acima descrito.

Belém, _____ de _____ de _____

Nome e assinatura do participante

Nome e assinatura do responsável por obter o consentimento

Testemunha

Testemunha

*Termo de Consentimento Livre Esclarecido apresentado, atendendo, conforme normas da Resolução 466/2012 de 12 de dezembro de 2012.

Apêndice IV. Imagens suplementares do aplicativo web, “MTS-Player”.

O Apêndice IV mostra algumas imagens do aplicativo web implementado para esta pesquisa. A linguagem de marcação HTML5 e a linguagem de programação Javascript foram utilizadas para a implementação desta ferramenta. Na Figura 2.2 exibe um arquivo .CSV inserido na área de inserção de tarefas. Na Figura AP-34 exibe-se uma tela de edição das tarefas. E na Figura AP-35, uma imagem de uma tarefa sendo executada pelo aplicativo.

Figura AP-33 – Área de inserção de tarefas do MTS-Player.



Figura AP-34 – Área de edição de tarefas do MTS-Player.

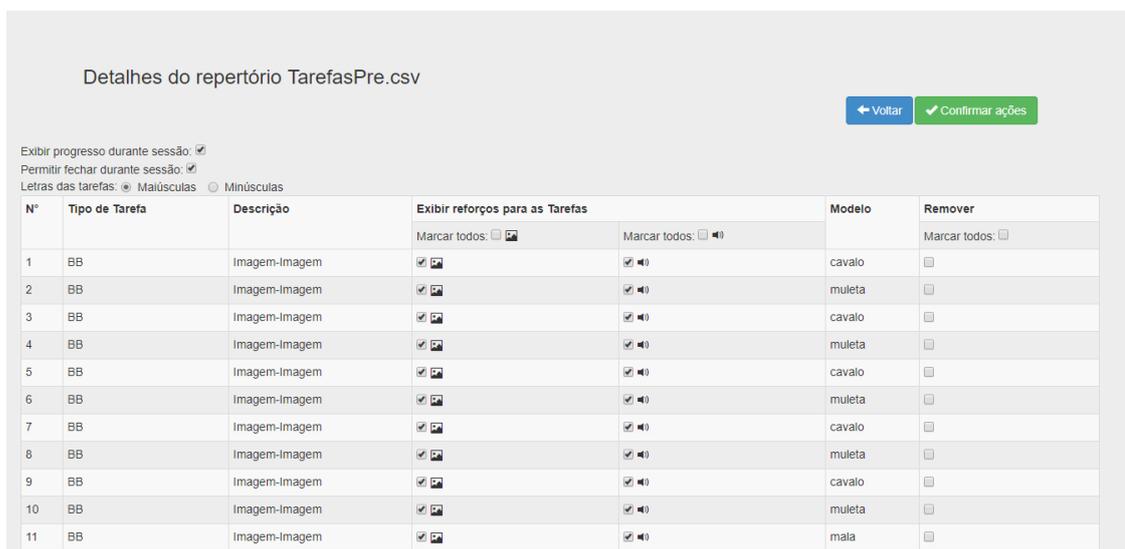


Figura AP-35 – Tarefa sendo apresentada no MTS-Player.



Apêndice V. Imagens suplementares do jogo digital, “As Aventuras de Amaru”.

O Apêndice V mostra algumas imagens do jogo digital implementado para esta pesquisa. As ferramentas usadas para implementação do jogo foram o Unity 3D, Gimp, Blender e linguagem de programação C#. Na Figura AP-36 exibe-se um recorte da história do jogo. Na Figura AP-37, um recorte do tutorial do jogo é exibido. A Figura AP-38 exibe os itens coletáveis durante o jogo e a Figura AP-39 exibe os “mini-games” do jogo.

Figura AP-36 – Início da história do jogo em formato de gravuras.



Figura AP-37 – Tutorial no início do jogo.



Figura AP-38 – Itens coletáveis entre as tarefas do jogo.

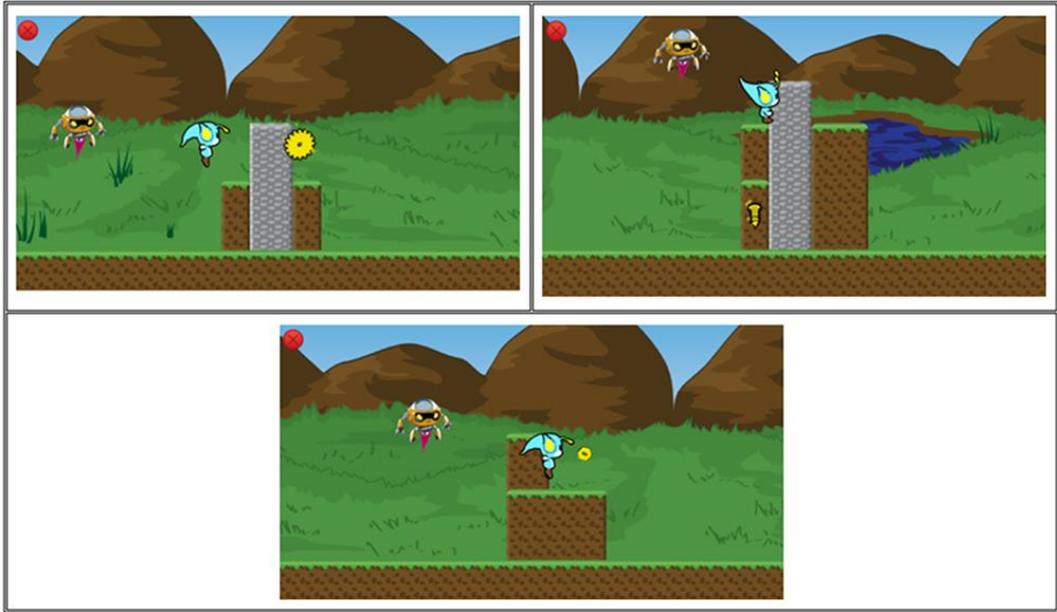
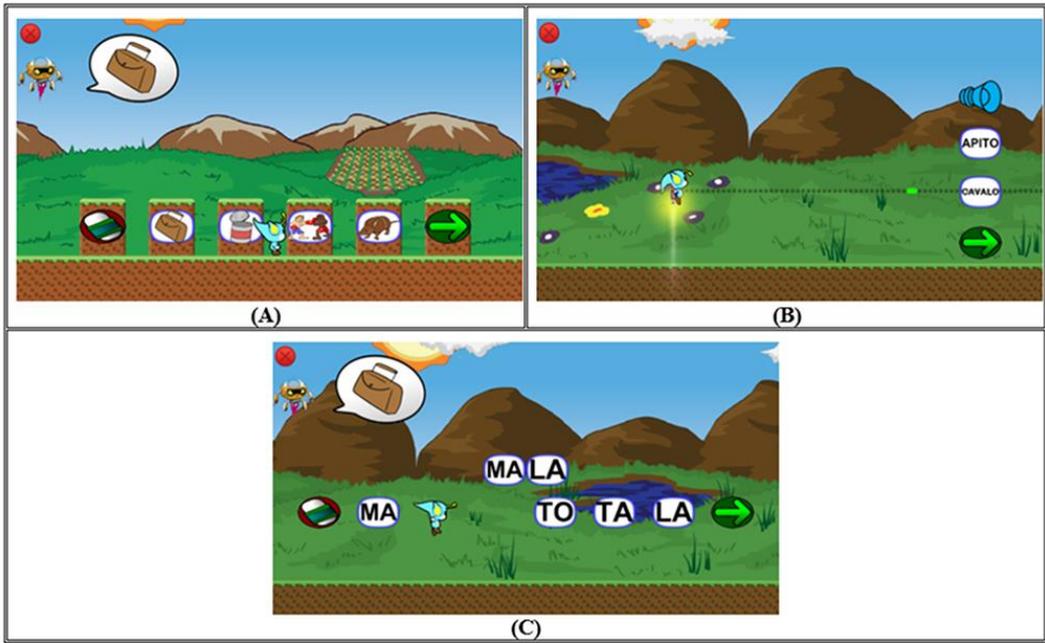


Figura AP-39 – Execução das tarefas em mini-games.



Apêndice VI. Questionário para validação das dificuldades das tarefas de ensino.

O Apêndice VI mostra algumas páginas do questionário como instrumento de coleta das opiniões dos especialistas. Os dados coletados serviram para identificar a dificuldade das tarefas de ensino e se as mesmas estavam adequadas para serem utilizadas com crianças em fase inicial da alfabetização. A seguir são apresentadas imagens que são pequenos recortes do questionário submetidos a especialistas educacionais.

Questionário: Dificuldade de tarefas de ensino (Versão Final)

Prezado(a), pesquisador(a), psicólogo(a) ou educador(a), convidamos você a colaborar com a realização da pesquisa intitulada "Geração automatizada de repertórios de ensinos baseado em fatores de dificuldade".

A sua participação envolve responder ao questionário que segue. Serão apresentadas tarefas de ensino de leitura informatizada com diferentes níveis de dificuldade. Pedimos que você categorize as tarefas de acordo com sua avaliação de dificuldade. As tarefas são direcionadas à crianças em fase inicial de alfabetização.

Esta pesquisa faz parte de um projeto de pesquisa do Laboratório de Inteligência Computacional e Pesquisa Operacional (LINC) – Instituto de Tecnologia (ITEC) da Universidade Federal do Pará (UFPA) e está sendo coordenado pelos professores Ádamo Santana (lattes: <http://goo.gl/YCR4m4>) e Dionne Monteiro (lattes: <http://goo.gl/QGF8wk>).

Todos os seus dados serão mantidos confidenciais e serão utilizados somente para os fins da pesquisa, mantendo sempre o seu nome em sigilo. Você pode se recusar a participar do estudo, ou retirar seu consentimento a qualquer momento, sem precisar justificar. Se você estiver de acordo com esse termo, nós gostaríamos que você preenchesse ao questionário a seguir. Isso não deve durar mais do que 10 minutos.

Ficaremos muito gratos por sua colaboração.

Qualquer dúvida, por favor, entre em contato conosco.

Gilberto Nerino Junior
<http://lattes.cnpq.br/8391942175575646>
Mestre em Ciência da Computação - PPGCC UFPA

*Obrigatório

Você aceita participar dessa pesquisa? *

- Sim
- Não

Você já trabalhou com o ensino da leitura ? *

- Sim
- Não

Qual sua experiência com trabalhos sobre alfabetização? *

- Muito experiente (mais de 7 anos)
- Experiente (entre 4 e 6 anos)
- Razoável experiência (entre 1 e 3 anos)
- Pouca experiência (menos de 1 ano)
- Nenhuma experiência

Caso queira receber mais informações sobre a pesquisa forneça seu endereço eletrônico

Quer saber mais sobre nosso trabalho? Por favor, informe seu email.

Qual a sua profissão? *

Caso você seja estudante de graduação, por favor, adicionar o nome do curso e o ano de término do curso no campo "Outro".

- Professor
- Pesquisador
- Psicólogo
- Pedagogo
- Terapeuta ocupacional

Outro:

Qual seu local de trabalho? *

- Universidade
- Clínica
- Hospital
- Escola

Outro:

Qual a sua faixa etária? *

- Dos 18 aos 22 anos
- Dos 23 aos 25 anos
- Dos 26 aos 30 anos
- Dos 31 aos 40 anos
- Dos 41 aos 50 anos
- Dos 51 aos 60 anos
- Maior de 61 anos

Gênero *

- Masculino
- Feminino

Instruções

Uma tarefa de ensino consiste na apresentação de uma tela composta por uma palavra modelo (ditada, figura ou texto impresso) e uma ou mais alternativas de escolha, que podem estar no formato de texto escrito ou figuras.

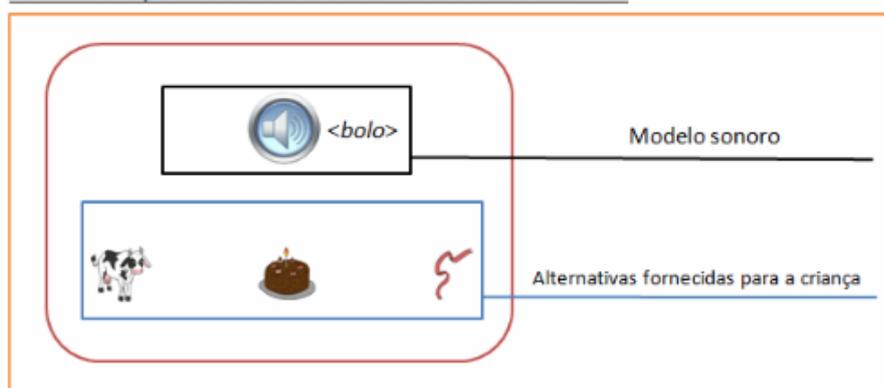
As figuras e textos impressos de modelo são sempre exibidos na parte superior da tela. A tarefa exige que a criança selecione dentre as alternativas de escolha disponíveis àquela que corresponde ao modelo. Em algumas tarefas a resposta consiste em selecionar, na ordem correta, as sílabas que compõem a palavra correspondente ao modelo

As figuras abaixo exemplificam algumas tarefas de ensino.

Observação:

As imagens que tem uma figura de um alto-falante com um texto ao lado representam estímulos sonoros. Ou seja, o som da palavra ditada por uma pessoa como modelo. O texto ao lado do alto-falante representa a palavra ditada.

Figura 1: A figura 1 mostra um ditado, ou seja, uma tarefa com um modelo sonoro da palavra "bolo". A criança deve selecionar a alternativa referente ao modelo.



As perguntas a seguir visam identificar o nível de dificuldade das tarefas que serão apresentadas a crianças em fase inicial de alfabetização. Presuma que as crianças não obtiveram treinamento com os tipos de tarefas de ensino e que estão nas séries iniciais de ensino da leitura.

Você deve indicar (com base na sua avaliação), o quão difícil será para as crianças responderem as tarefas que seguem.

As tarefas serão apresentadas todas em conjunto. Isso permitirá que você possa, a qualquer momento, reavaliar a sua classificação de cada tarefa.

tomate



5 - Qual o nível de dificuldade da tarefa de ensino (exibida acima) para crianças em fase inicial de alfabetização? *

- 1 - Muito fácil
- 2 - Fácil
- 3 - Média
- 4 - Difícil
- 5 - Muito difícil



vaca

tatu

8 - Qual o nível de dificuldade da tarefa de ensino (exibida acima) para crianças em fase inicial de alfabetização? *

- 1 - Muito fácil
- 2 - Fácil
- 3 - Média
- 4 - Difícil
- 5 - Muito difícil



<tomate>



33 - Qual o nível de dificuldade da tarefa de ensino (exibida acima) para crianças em fase inicial de alfabetização? *

- 1 - Muito fácil
- 2 - Fácil
- 3 - Média
- 4 - Difícil
- 5 - Muito difícil

Anexos

Esta seção tem como objetivo mostrar documentos e informações complementares desenvolvidos por terceiros a esta pesquisa.

Anexo A. Parecer Consubstanciado do CEP (Comitê de Ética e Pesquisa)

INSTITUTO DE CIÊNCIAS DA
SAÚDE DA UNIVERSIDADE
FEDERAL DO PARÁ - ICS/



PARECER CONSUBSTANCIADO DO CEP

DADOS DO PROJETO DE PESQUISA

Título da Pesquisa: Jogos computacionais para o processo de ensino-aprendizagem de crianças em fase de alfabetização.

Pesquisador: GILBERTO NERINO DE SOUZA JUNIOR

Área Temática:

Versão: 1

CAAE: 39059214.3.0000.0018

Instituição Proponente:

Patrocinador Principal: FUNDACAO AMAZONIA PARAENSE DE AMPARO A PESQUISA - FAPESPA

DADOS DO PARECER

Número do Parecer: 978.783

Data da Relatoria: 26/01/2015

Apresentação do Projeto:

Trata-se de um estudo que avaliará a utilização de jogos computacionais no processo de ensino-aprendizagem de crianças em fase de alfabetização.

Objetivo da Pesquisa:

O Objetivo do experimento é medir o engajamento nos jogos desenvolvidos em contextos de aprendizado de leitura e soletração. De modo mais específico, o objetivo primário deste projeto visa desenvolver duas ferramentas de ensino de leitura e escrita em português brasileiro, para o processo de ensino-aprendizagem de crianças, baseado em jogos computacionais e no programa de ensino ALEPP, utilizando para isso mecanismos de interação lúdicos.

Avaliação dos Riscos e Benefícios:

O estudo apresenta riscos mínimos, pretendendo maximizar os benefícios quando proporcionar um ambiente de ensino-aprendizagem mais lúdico e eficiente, em comparação às estratégias de ensino formais tornando-o mais motivador e eficaz.

Endereço: Rua Augusto Corrêa nº 01-SI do ICS 13 - 2º and.

Bairro: Campus Universitário do Guamá **CEP:** 66.075-110

UF: PA **Município:** BELEM

Telefone: (91)3201-7735

Fax: (91)3201-8028

E-mail: cepccs@ufpa.br

INSTITUTO DE CIÊNCIAS DA
SAÚDE DA UNIVERSIDADE
FEDERAL DO PARÁ - ICS/



Continuação do Parecer: 978.783

Comentários e Considerações sobre a Pesquisa:

O estudo mostra-se de grande relevância para o processo de ensino na alfabetização.

Considerações sobre os Termos de apresentação obrigatória:

Apresenta todos os documentos necessários para a elaboração do parecer e conseqüente aprovação da pesquisa.

Recomendações:

Sem recomendações.

Conclusões ou Pendências e Lista de Inadequações:

Apto para aprovação.

Situação do Parecer:

Aprovado

Necessita Apreciação da CONEP:

Não

Considerações Finais a critério do CEP:

BELEM, 10 de Março de 2015

Assinado por:
Wallace Raimundo Araujo dos Santos
(Coordenador)

Endereço: Rua Augusto Corrêa nº 01-SI do ICS 13 - 2º and.

Bairro: Campus Universitário do Guamá **CEP:** 66.075-110

UF: PA **Município:** BELEM

Telefone: (91)3201-7735 **Fax:** (91)3201-8028 **E-mail:** cepccs@ufpa.br