

JOÃO GABRIEL GAMA VILA NOVA

APRENDENDO CURRÍCULOS PARA HUMANOS COM BUSCA EM  
ÁRVORE GUIADA POR REDE NEURAL

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Magister Scientiae*.

Orientador: Levi Henrique Santana de Lelis

VIÇOSA - MINAS GERAIS  
2021

**Ficha catalográfica elaborada pela Biblioteca Central da Universidade  
Federal de Viçosa - Campus Viçosa**

T

N935a  
2021  
Nova, João Gabriel Gama Vila, 1995-  
Aprendendo currículos para humanos com busca em árvore  
guiada por rede neural / João Gabriel Gama Vila Nova. – Viçosa,  
MG, 2021.

1 dissertação eletrônica (52 f.): il. (algumas color.).

Inclui apêndice.

Orientador: Levi Henrique Santana de Lelis.

Dissertação (mestrado) - Universidade Federal de Viçosa,  
Departamento de Informática, 2021.

Referências bibliográficas: f. 36-38.

DOI: <https://doi.org/10.47328/ufvbbt.2022.030>

Modo de acesso: World Wide Web.

1. Estrutura de dados (Computação). 2. Educação. 3. Jogos.  
I. Lelis, Levi Henrique Santana de, 1984-. II. Universidade  
Federal de Viçosa. Departamento de Informática. Programa de  
Pós-Graduação em Ciência da Computação. III. Título.

CDD 22. ed. 005.73

Bibliotecário(a) responsável: Renata de Fátima Alves CRB6/2578

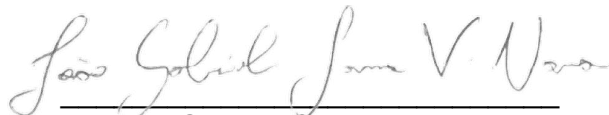
JOÃO GABRIEL GAMA VILA NOVA

**APRENDENDO CURRÍCULOS PARA HUMANOS COM BUSCA EM  
ÁRVORE GUIADA POR REDE NEURAL**

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Magister Scientiae*.

APROVADA: 17 de dezembro de 2021.

Assentimento:



---

João Gabriel Gama Vila Nova  
Autor



---

Levi Henrique Santana de Lelis  
Orientador

# Agradecimentos

Agradeço primeiro a minha família que me deu todo apoio para trilhar este caminho longe de casa e me proporcionou a base para que pudesse caminhar com minhas próprias pernas. Agradeço também a Larissa e Amora que vieram juntas transformar um pequeno apartamento em um verdadeiro lar. Aos amigos, velhos e novos, que mesmo distante nunca estiveram tão presentes em uma época tão difícil e cheia de restrições. E por último ao meu orientador, Levi, que não só me orientou mas também me apoiou durante toda essa trajetória.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

# Resumo

VILA NOVA, João Gabriel Gama, M.Sc., Universidade Federal de Viçosa, dezembro de 2021. **Aprendendo Currículos para Humanos com Busca em Árvore Guiada por Rede Neural.** Orientador: Levi Henrique Santana de Lelis

Este trabalho avalia Levin Tree Search (LTS) como um modelo para aprender currículos para pessoas. Nós hipotetizamos que a ordem em que LTS aprende a resolver um conjunto de problemas pode ser usada para formar um currículo que auxilia o aprendizado de humanos. Nós avaliamos LTS em dois esquemas de aprendizagem de currículos. Primeiro, ordenamos um pequeno conjunto de instâncias de problemas de acordo com a ordem em que LTS as resolve enquanto aprende uma política neural. Segundo, usamos LTS para ordenar um grande conjunto de instâncias de problemas e selecionamos um subconjunto deste para formar um currículo. Nós avaliamos o currículo que LTS gera através de um estudo com usuários em que os participantes aprendem a resolver uma classe de puzzles do jogo The Witness. Resultados computacionais mostram que a ordem que LTS encontra para o conjunto de instâncias do jogo é a mesma ordem escolhida pelo game designer do jogo. O resultado do estudo com usuários sugere que o currículo que nosso sistema gera se compara favoravelmente com currículos de base em termos de retenção de usuário e número de tentativas para que se consiga resolver uma instância. Nossos experimentos computacionais e estudos com usuários sugerem que LTS utilizando uma política neural pode ser usada como um efetivo modelo de geração de currículos de ensino para auxiliar o aprendizado de pessoas em problemas de decisão sequencial.

**Palavras-chave:** Educação. Busca em Árvore. Jogos.

# Abstract

VILA NOVA, João Gabriel Gama, M.Sc., Universidade Federal de Viçosa, December, 2021. **Learning Curricula for Humans with Neural-Guided Tree Search**. Advisor: Levi Henrique Santana de Lelis.

This work evaluates Levin tree search (LTS) as a model for learning curricula for humans. We hypothesize the order in which LTS learns how to solve a set of problem instances can be used to form a curriculum for helping human learners. We evaluate LTS in two curriculum learning settings. First, we order a small set of problem instances according to the order in which LTS solves them while learning a neural policy. Second, we use LTS to order a large set of problem instances and we select a subset of instances to form a curriculum. We evaluate the curricula LTS generates with a user study where participants learn how to solve a class of puzzles from the commercial game The Witness. Computational results show the ordering LTS finds for the small set of instances matches the ordering chosen by a professional game designer. The user study results suggest the curricula our system generates compare favorably with baseline curricula in terms of user retention and number of attempts required to solve test instances. Our computational experiments and the user study suggest that LTS employing a neural policy can be used as an effective curriculum learning model to aid human learning in sequential decision-making settings.

**Keywords:** Education. Tree Search. Games.

# Lista de ilustrações

Figura 1 – Árvore de estados de um problema de decisão sequencial de um único agente. O estado inicial $s_0$ se ramifica a cada ação que o altera, os valores acima de cada nó são as probabilidades para a escolha da ação que gera o estado dadas por uma política ( $\pi$ ). . . . .	17
Figura 2 – Currículo original do jogo ‘The Witness.’ O jogador soluciona as instâncias da esquerda para a direita. . . . .	24
Figura 3 – As instâncias de teste usadas em nosso estudo. A primeira instância é advinda do jogo e é fixada para todos os participantes. As outras duas são apenas representativas das instâncias de teste. . . . .	26
Figura 4 – A solução de nosso sistema para o problema da ordenação e seleção (currículo equidistante). . . . .	26
Figura 5 – Exemplo de currículo aleatório. . . . .	26
Figura 6 – Formulário de consentimento. . . . .	47
Figura 7 – Questionário demográfico. . . . .	48
Figura 8 – Instruções em como usar os controles do teclado. . . . .	49
Figura 9 – Instruções finais antes de começar a resolver as instâncias do currículo. . . . .	49
Figura 10 – Interface de um dos puzzles, com alguns puzzles já resolvidos aparecendo no topo. . . . .	50
Figura 11 – Interface da página de feedback geral para nosso experimento. . . . .	51
Figura 12 – Página onde damos Feedback ao participante. . . . .	52

# Lista de tabelas

Tabela 1	– A taxa de desistência e o número de <i>puzzles</i> resolvidos, como M (DP), por participante antes de desistir. Letras sobrescritas ( <i>a, b, c, d, e</i> ) significam diferentes distribuições no teste pareado de Fisher e Mann-Whitney para a primeira e segunda coluna respectivamente. . . . .	30
Tabela 2	– Número de <i>puzzles</i> solucionados como M (DP). Letras sobrescritas ( <i>a, b</i> ) significam diferentes distribuições no teste pareado de Mann-Whitney, usado apenas entre os três currículos do witness. . . . .	30
Tabela 3	– Número de tentativas, como M (DP), para as instâncias do currículo, de teste e para a primeira instância de teste (p10). Letras sobrescritas ( <i>a, b</i> ) significam diferentes distribuições no teste pareado Mann-Whitney.	31
Tabela 4	– Número de <i>backtracks</i> , como M (DP), por tipo de instância e currículo. Letras sobrescritas ( <i>a, b</i> ) significam diferentes distribuições no teste pareado Mann-Whitney. . . . .	32
Tabela 5	– Tempo, como M (DP), por tipo de instância e currículo. . . . .	32
Tabela 6	– Resultados do teste Exato de Fisher entre cada um dos pares de currículos para a taxa de desistência. . . . .	41
Tabela 7	– Resultados dos testes de Mann-Whitney entre os pares de currículos para o número de <i>puzzles</i> resolvidos antes de desistir do experimento.	41
Tabela 8	– Resultados dos testes de Mann-Whitney entre os pares de currículos para o número de <i>puzzles</i> que os participantes não-desistentes resolveram. . . . .	41
Tabela 9	– Fração dos participantes não-desistentes que resolveu cada <i>puzzle</i> por currículo. . . . .	42
Tabela 10	– Resultados dos testes de Mann-Whitney entre os pares de currículos para o número de tentativas nos <i>puzzles</i> de teste que os participantes não-desistentes performaram. . . . .	42
Tabela 11	– Número de tentativas apresentado como M (DP) por <i>puzzles</i> e currículo. . . . .	42
Tabela 12	– Resultados dos testes de Mann-Whitney entre os pares de currículos para o número de <i>backtracks</i> nos <i>puzzles</i> dos currículos que os participantes não-desistentes performaram. . . . .	43
Tabela 13	– Número de <i>backtracks</i> apresentados como M (DP) por <i>puzzle</i> e currículo. . . . .	43
Tabela 14	– Número de participantes por país e currículo antes de contar os desistentes. . . . .	43

Tabela 15 – Número de participantes por país e currículo depois de contar os desistentes. . . . .	44
Tabela 16 – Número de participantes por gênero e currículo antes de contar os desistentes. . . . .	44
Tabela 17 – Número de participantes por gênero e currículo depois de contar os desistentes. . . . .	44
Tabela 18 – Número de participantes por idade e currículo antes de contar os desistentes. . . . .	45
Tabela 19 – Número de participantes por idade e currículo depois de contar os desistentes. . . . .	45
Tabela 20 – Número de participantes por nível de escolaridade e currículo antes de contar os desistentes. . . . .	45
Tabela 21 – Número de participantes por nível de escolaridade e currículo depois de contar os desistentes. . . . .	45
Tabela 22 – Número de participantes por frequência de jogar jogos e currículo antes de contar os desistentes. . . . .	46
Tabela 23 – Número de participantes por frequência de jogar jogos e currículo depois de contar os desistentes. . . . .	46
Tabela 24 – Número de participantes por frequência de resolver puzzles e currículo antes de contar os desistentes. . . . .	46
Tabela 25 – Número de participantes por frequência de resolver puzzles e currículo depois de contar os desistentes. . . . .	46

# Sumário

<b>1</b>	<b>Introdução</b>	<b>11</b>
1.1	Objetivos	11
1.1.1	Objetivos Específicos	12
1.2	Hipótese	12
1.3	Contribuição	12
1.4	Estrutura da Dissertação	13
<b>2</b>	<b>Trabalhos Relacionados</b>	<b>14</b>
2.1	Computer-Aided Instruction	14
2.2	Machine Teaching Direcionado à Algoritmos	14
2.3	Machine Teaching Direcionado aos Humanos	15
<b>3</b>	<b>Definição do Problema</b>	<b>16</b>
<b>4</b>	<b>Levin Tree Search (LTS)</b>	<b>17</b>
<b>5</b>	<b>O Algoritmo de Bootstrap</b>	<b>19</b>
<b>6</b>	<b>Aprendizagem de Currículo com Bootstrap</b>	<b>21</b>
6.1	Problema de Ordenação	21
6.2	Problema de Ordenação e Seleção	21
<b>7</b>	<b>Desenho Experimental</b>	<b>23</b>
7.1	Black and White Square puzzles	23
7.2	Treinando o Modelo Neural	24
7.3	Currículos para o Estudo	25
7.4	Instâncias de Teste para o Estudo	26
7.5	Métricas	27
7.6	Testes Estatísticos	27
<b>8</b>	<b>Resultados do Estudo com Usuários</b>	<b>29</b>
8.1	Retenção de Usuário	29
8.2	Número de Instâncias Resolvidas	30
8.3	Esforço dos Usuários	31
8.3.1	Tentativas	31
8.3.2	Backtracks	32
8.3.3	Tempo	33
8.4	Discussão	33
<b>9</b>	<b>Conclusão</b>	<b>35</b>
	<b>Referências</b>	<b>36</b>

<b>Apêndices</b>	<b>39</b>
<b>APÊNDICE A Material Suplementar</b>	<b>40</b>
A.1 Retenção de Usuário	40
A.2 Número de Instâncias Resolvidas	40
A.3 Esforço dos Usuários	40
A.4 Dados Demográficos	43
A.5 Interface Gráfica do Usuário	44

# 1 Introdução

A combinação de busca em árvore e redes neurais alcançou performance sobre-humana em domínios complexos como Xadrez e Go (SILVER et al., 2018). Tendo em vista o potencial destes modelos e a possibilidade que existe em se aprender com os agentes resultantes dos mesmos, estamos interessados em gerar um currículo para pessoas que capture o conhecimento que esses sistemas inteligentes são capazes de gerar. Usamos um algoritmo de busca em árvore guiado por uma rede neural para gerar currículos para aprendizes humanos (estudantes) no contexto de problemas de decisão sequencial de um único agente.

Um currículo é um conjunto ordenado de instâncias de problemas que é criado para facilitar o aprendizado. Estudantes começam por tentar resolver a primeira instância no currículo e então avançam para a segunda sequencialmente. Um currículo bem feito gradualmente introduz novos conceitos para que conceitos aprendidos durante a resolução de instâncias anteriores sejam utilizados para dar suporte ao aprendizado posterior. Um currículo deve introduzir novos conceitos no momento certo já que estudantes podem ficar entediados se as instâncias apresentadas forem muito fáceis ou podem ficar frustrados caso estas sejam difíceis demais (GRAESSER; D'MELLO, 2012). Consequentemente, introduzir novos conceitos no momento errado pode resultar em aumento das taxas de desistência. Finalmente, currículos deveriam cobrir todos os conceitos necessários para resolver qualquer instância de uma dada classe de problemas.

Utilizamos como nossa classe de problemas de decisão sequencial de um único agente *puzzles* do jogo The Witness. Escolhemos uma classe específica dos *puzzles* desse jogo pois no The Witness os jogadores são apresentados às instâncias sem que sejam ensinados como resolvê-las. O jogo faz então um bom trabalho de selecionar as instâncias em uma ordem certa para que os jogadores consigam aprender a solucionar os problemas a partir de tentativa e erro. Uma classe de problemas com essa característica é interessante para o objetivo deste trabalho já que ele envolve diretamente o ensino de conceitos para aprendizes.

## 1.1 Objetivos

O objetivo principal deste trabalho é a geração de um currículo eficiente voltado ao preparo de estudantes para a resolução de problemas de decisão sequencial de um único agente. Usamos como suporte para nossos testes */puzzles* do jogo The Witness.

### 1.1.1 Objetivos Específicos

- Desenvolver um modelo baseado em busca em árvore e redes neurais que seja capaz de aprender a solucionar problemas da classe proposta.
- Desenvolver um gerador de problemas da classe de *puzzles*.
- Testar a capacidade do modelo de ordenar as instâncias de *puzzles* em uma ordenação coerente com o aprendizado.
- Gerar um currículo a partir dos resultados do aprendizado do modelo com as instâncias de *puzzles* que seja efetivo no ensino dessa classe de problemas.
- Experimentar o currículo gerado em um experimento com usuários.

## 1.2 Hipótese

Nossa hipótese é que a ordem em que LTS resolve um conjunto de instâncias enquanto aprende uma política gerada por uma rede neural pode ser utilizada para gerar um currículo efetivo para estudantes.

## 1.3 Contribuição

Neste trabalho, investigamos *Levin Tree Search* (LTS) (ORSEAU et al., 2018) e o sistema de aprendizagem *Bootstrap* (ARFAEE; ZILLES; HOLTE, 2011) como um modelo para aprender currículos para problemas de decisão sequencial. Nós avaliamos nossa hipótese com dois esquemas. O primeiro ordena um conjunto de instâncias para formar um currículo (**problema de ordenação**). O segundo utiliza um conjunto maior de instâncias e seleciona um subconjunto ordenado dessas instâncias para formar um currículo (**problema de ordenação e seleção**).

Nós testamos nossa hipótese através de um estudo com usuários usando uma classe de *puzzles* do jogo The Witness. Conduzimos nosso estudo pela plataforma Lab in the Wild (REINECKE; GAJOS, 2015) por mais de 2 anos, obtendo 607 participantes. Dos cinco currículos testados, dois foram gerados automaticamente por nosso sistema. Um desses currículos é a solução do problema de ordenação quando provemos um pequeno conjunto de *puzzles* do jogo original como entrada. O segundo é a solução do problema de ordenação e seleção, onde usamos um grande conjunto de *puzzles* gerados para selecionar deles um subconjunto. Os resultados do estudo com usuários suportam nossa hipótese: O currículo gerado por nosso sistema se compara favoravelmente com outros currículos de base em termos de retenção e esforço (e.g., o número de tentativas requeridas para se resolver um conjunto de *puzzles* de teste). Suporte adicional à nossa hipótese vem

da ordenação gerada no problema de ordenação que se iguala a ordenação das mesmas instâncias no jogo original, que foi criada por um experiente game designer. Além disso, nosso currículo gerado pelo problema da ordenação e seleção inclui um *puzzle* com uma característica rara que o torna difícil tanto para as pessoas quanto para a máquina. Isso sugere que nosso modelo é capaz de capturar conceitos importantes, mesmo que eles sejam elusivos.

## 1.4 Estrutura da Dissertação

Essa dissertação está dividida da seguinte maneira: no capítulo 2 é feita uma breve revisão de trabalhos relacionados ao projeto. O capítulo 3 apresenta a definição dos problemas que serão abordados nos testes. No capítulo 4 e 5 são apresentadas as principais técnicas utilizadas em nosso modelo. No capítulo 6 é apresentado o desenho do experimento rodado online, com todos os seus grupos e métricas, cujos resultados são analisados no capítulo 7. Por fim, o capítulo 8 apresenta as conclusões que obtivemos e sugestões para trabalhos futuros.

## 2 Trabalhos Relacionados

Nosso trabalho é relacionado ao uso de software para auxiliar estudantes, por exemplo, *Intelligent Tutoring Systems* (ITS) (VANLEHN, 2006) e *Computer-Aided Instruction* (CAI). Também é relacionado com *Machine Teaching* (ZHU, 2015; ZHU et al., 2018), onde se está interessado em gerar um conjunto de treinamento que permita que um estudante aprenda um conceito de maneira eficiente (ex: limite de decisão de um classificador).

### 2.1 Computer-Aided Instruction

ITSs são um tipo de CAI que se adapta para cada estudante em nível individual (VANLEHN, 2006; FENZA; ORCIUOLI; SAMPSON, 2017). A fundação histórica dos ITSs em sistemas especialistas significa que eles necessitam de um considerável esforço humano para se tornarem adaptativos. Uma pessoa precisa criar um currículo base e desenvolver adaptações apropriadas na ausência de dados históricos de uso. Os ITSs datam dos anos 1970s (DOROUDI; ALEVEN; BRUNSKILL, 2019) e têm sido usados para apoiar o ensino através de uma variedade de domínios (RUNZRAT; HARFIELD; CHAROEN-SIRIWATH, 2019; CORBETT et al., 2006; BOSCH et al., 2015). Quando uma coletânea histórica de interação de estudantes existe, pessoas têm simulado interações para testar versões modificadas de currículos (MATSUDA et al., 2007) ou construir modelos que melhor adaptam currículos para estudantes enquanto suas habilidades mudam (PARDOS; DADU, 2018).

Em um estudo bibliográfico sobre pesquisas em sequenciamento de currículo para aquisição de conceitos, nenhuma diferença consistente em ganho de conhecimento dos estudantes foi encontrada quando comparando currículos criados por humanos com aqueles criados usando aprendizagem por reforço (DOROUDI; ALEVEN; BRUNSKILL, 2019). Nós visamos apoiar este mesmo tipo de aprendizado. Diferente dos outros, nós o fizemos sem acesso a dados históricos de interações. Nós ainda vamos além desse paradigma usando algoritmos para gerar a sequência de conteúdo inicial que foi historicamente criada por profissionais do domínio.

### 2.2 Machine Teaching Direcionado à Algoritmos

Um dos desafios em *machine teaching* é a geração de currículos, que é a tarefa de criar um pequeno conjunto de problemas que seja suficiente para ensinar um conceito. Normalmente o problema da geração de currículos é atacado começando com instân-

cias fáceis e gradualmente aumentando a dificuldade das mesmas (BENGIO et al., 2009; TAYLOR, 2009). Outros tentaram resolver *machine teaching* por meio de uma percepção teórica (GOLDMAN; KEARNS, 1995; KOBAYASHI; SHINOHARA, 2009; ZILLES et al., 2011; ZHU; LIU; LOPES, 2017). Os trabalhos acima focam na eficiência da seleção da amostra. O sistema que avaliamos encontra uma aproximação do tamanho ótimo de conjunto de treinamento para um algoritmo de aprendizado, o que é parecido com o objetivo desses trabalhos anteriores. No entanto, nós avaliamos um subconjunto deste quase ótimo conjunto de treinamento como um currículo para pessoas.

## 2.3 Machine Teaching Direcionado aos Humanos

Muitos estudos modelaram a tarefa de escolher a próxima atividade para um estudante como um problema de planejamento (WHITEHILL, 2009; CHI et al., 2011; RAFFERTY et al., 2011) ou como um problema de otimização (PATIL et al., 2014; LINDSEY et al., 2013). Esses métodos observam como o estudante se comporta antes de decidir a próxima atividade. O nosso sistema desenvolve um currículo previamente e não observamos o estudante. Comumente, métodos baseados em planejamento e otimização assumem a existência de um modelo de estudante enquanto nós avaliamos um algoritmo de busca em árvore guiado por uma rede neural como nosso modelo de estudante. Esses trabalhos foram avaliados com tarefas de classificação, enquanto nós focamos em problemas de decisão sequencial.

Outros avaliam as estratégias que pessoas usam para ensinar (KHAN; MUTLU; ZHU, 2011; CAKMAK; LOPES, 2012). Os resultados de Khan, Mutlu e Zhu mostram que professores humanos usam uma estratégia similar ao modelo de aprendizagem de currículo já que estudantes tendem a ser expostos primeiramente a problemas fáceis e apenas depois aos mais difíceis. Como este trabalho, nós comparamos o currículo que nosso sistema gera com o de um professor humano. Enquanto Khan, Mutlu e Zhu focam novamente em tarefas de classificação, nós focamos nas de decisão sequencial. Cakmak e Lopes também trabalham com decisão sequencial, mas eles focam em treinar professores humanos enquanto nós focamos no ensino de estudantes humanos.

### 3 Definição do Problema

Seja  $P = (S, A, T, s_0, g)$  um problema de planejamento determinístico, onde  $S$  é um conjunto de estados e  $s_0$  e  $g$  em  $S$  sejam o estado inicial e o objetivo.  $A$  é a função que recebe um estado  $s$  e retorna um conjunto de ações válidas  $a$  em  $s$ .  $T$  é a função de transição determinística que recebe o estado  $s$  e uma ação  $a$  e retorna o estado alcançado quando  $a$  é aplicado a  $s$ . Um caminho de solução é uma sequência de pares estado-ação  $\{(s_i, a_i)\}_{i=0}^m$  em que  $T(s_i, a_i) = s_{i+1}$  e  $T(s_m, a_m) = g$ . Uma instância de problema é definida como  $(s_0, g)$  para  $S, A,$  e  $T$  fixos.

Um algoritmo de aprendizagem  $L(D)$  recebe um conjunto ordenado de instâncias de problemas  $D$  de um problema  $P$  e um algoritmo de busca e retorna uma política  $\pi^*$ , por exemplo, a política que minimiza o número de estados que o algoritmo de busca guiado por  $\pi^*$  deve visitar para encontrar soluções para as instâncias de  $P$ . A política  $\pi(s, a)$  retorna a probabilidade de tomar a ação  $a$  em  $s$ .

A eficiência de  $L$  é o número de estados visitados enquanto aprende  $\pi^*$ , que depende da ordenação de  $D$ , como explicaremos depois. A função  $L^{-1}$  recebe  $\pi$  e retorna um conjunto ordenado  $D^*$  que minimiza o número de estados que  $L$  precisa visitar para aprender  $\pi^*$ .

Em aprendizagem de currículos para problemas de planejamento, recebe-se um conjunto de instâncias  $D$  em uma ordenação arbitrária e retorna-se um conjunto ordenado  $D^*$  que minimiza o número de estados que  $L$  visita para aprender  $\pi^*$ . Nós hipotetizamos que o  $D^*$  que  $L^{-1}$  retorna pode ser usado como um currículo efetivo para estudantes humanos.

Em resumo, nosso problema é: Dado um conjunto de instâncias  $D$ , queremos gerar um conjunto ordenado  $D^*$  onde sua ordenação seja positiva para o aprendizado de estudantes da classe de problemas das instâncias.

Nós usamos *Bootstrap* e *LTS* para aproximar  $\pi^*$ . Nas próximas seções, revisaremos estes algoritmos e explicaremos como eles são usados para aproximar a função  $L^{-1}$ .

## 4 Levin Tree Search (LTS)

LTS é um algoritmo de busca do tipo ‘melhor-primeiro’ que usa uma política para resolver problemas de planejamento. LTS cria uma árvore de busca em que cada nó representa um estado em  $S$  e sua raiz é  $s_0$ . Já que cada nó representa um estado, usamos as palavras ‘nó’ e ‘estado’ intercambiavelmente. Existe uma aresta direta de  $n$  para  $n'$  na árvore se  $T(n, a) = n'$  com  $a \in A(n)$ . Um nó  $n$  é expandido quando computamos  $T(n, a)$  para todo  $a$  em  $A(n)$ . Nós obtidos com  $T(n, a)$  são chamados de filhos de  $n$  e falamos que os filhos são gerados quando seus pais são expandidos.

Como o algoritmo A\* (HART; NILSSON; RAPHAEL, 1968), LTS usa uma fila de prioridade, a qual nos referimos como OPEN. LTS começa inserindo o nó que representa o estado inicial  $s_0$  em OPEN. A cada iteração, ele remove de OPEN e expande o nó  $n$  com o menor custo. A função de custo que a LTS utiliza é  $\frac{d(n)}{\pi(n)}$ , denominado Custo Levin. Onde,  $d(n)$  é a profundidade de  $n$  na árvore de busca e  $\pi(n)$  é a probabilidade de  $n$ , que é computada como segue. Seja  $\{(n_i, a_i)\}_{i=0}^m$  a sequência de pares estado-ação começando da raiz da árvore  $n_0$  e terminando no nó  $T(n_m, a_m) = n$ , então  $\pi(n) = \prod_{i=0}^m \pi(n_i, a_i)$ . Essa função de custo permite que LTS foque em nós  $n$  com maiores valores de  $\pi(n)$ , enquanto  $d(n)$  age como um termo de exploração, forçando a busca a balancear os valores de probabilidade e profundidade. LTS conclui a busca assim que gera um nó solução ou quando OPEN fica vazia.

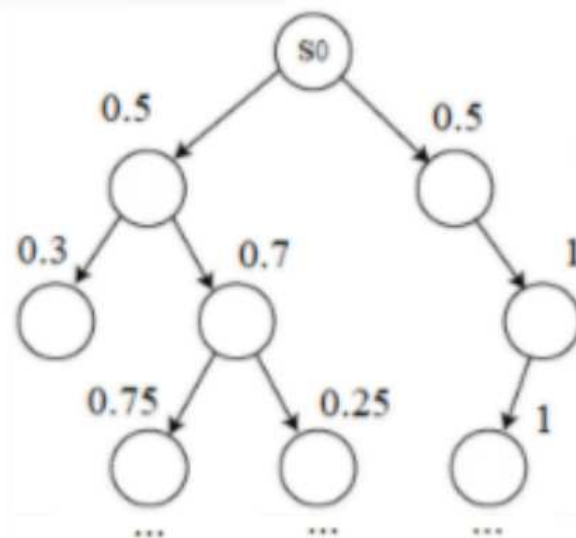


Figura 1 – Árvore de estados de um problema de decisão sequencial de um único agente. O estado inicial  $s_0$  se ramifica a cada ação que o altera, os valores acima de cada nó são as probabilidades para a escolha da ação que gera o estado dadas por uma política ( $\pi$ ).

Na Figura 1 temos um exemplo de árvore de estados, onde cada estado é alcan-

çado ao fazer uma ação a partir do estado anterior. Os números na figura representam a probabilidade de escolha de determinada ação que gera o próximo estado de acordo com a política ( $\pi$ ). Olhando para esta imagem podemos observar que caso não houvesse um termo de exploração na fórmula do Custo Levin ( $\frac{d(n)}{\pi(n)}$ ), representado nesta como a profundidade ( $d$ ), a busca poderia se perder indefinidamente no ramo da direita assumindo que todos os estados tivessem apenas um filho com a probabilidade 1. Para não sofrer deste problema de exploração a análise da profundidade dos nós faz com que quanto mais se aprofunde a busca em um ramo, mais esse ramo se torna desinteressante para a fila de prioridade em OPEN. Sendo assim, o termo  $d$  balanceia a exploração entre o que a política  $\pi$  aponta e a profundidade dos nós expandidos.

## 5 O Algoritmo de Bootstrap

O *Bootstrap* é um procedimento iterativo que usa a aprendizagem para criar uma série de funções heurísticas (ARFAEE; ZILLES; HOLTE, 2011). Ele recebe como entrada um conjunto de instâncias  $D$  e um algoritmo de busca  $R$ . Ele retorna uma política  $\pi$ . *Bootstrap* inicia com uma política  $\pi_0$  dada por uma rede neural inicializada aleatoriamente e aplica  $R$  guiado por  $\pi_0$  para tentar solucionar as instâncias em  $D$ ; as instâncias que  $R$  consegue solucionar, denotadas como  $V_0$ , são usadas para treinar a política, assim gerando  $\pi_1$ . O processo é repetido com  $\pi_1$  para as instâncias que não foram resolvidas na primeira iteração. As instâncias que  $R$  conseguir resolver com  $\pi_1$  formam o subconjunto  $V_1$  de  $D$ . A política  $\pi_2$  é obtida treinando  $\pi_1$  com  $V_0 \cup V_1$ . *Bootstrap* conclui assim que resolve todas as instâncias,  $\bigcup_i V_i = D$ , e retorna a última política  $\pi_n$  como uma aproximação de  $\pi^*$ .

*Bootstrap* usa um cronograma para ajustar a quota computacional (*budget*)  $b$  permitida para cada instância. A *budget* de busca é medida em termos de estados que a LTS visita durante a busca e é fixa por toda uma iteração (ex: LTS tenta resolver todas as instâncias em  $D$  enquanto visita no máximo  $b$  estados por instância). A *budget* é setada empiricamente para minimizar o esforço de busca de uma política em aprendizado. Por exemplo, Orseau e Lelis (2021) usaram um  $b$  inicial de 2,000 para os *puzzles* do The Witness que usamos em nossos experimentos. Se  $R$  não é capaz de resolver nenhuma instância em uma dada iteração, o *Bootstrap* dobra  $b$  para a próxima iteração.

O conjunto parcialmente ordenado  $V = \{V_0, V_1, \dots, V_n\}$  aproxima  $L^{-1}(\pi_n)$  porque ele permite que se aprenda  $\pi_n$  apenas com as instâncias que são solucionadas em  $D$ . Isto é,  $R$  resolve as instâncias em  $V_0$  com  $\pi_0$ , que provê a política atualizada  $\pi_1$ .  $R$  então usa  $\pi_1$  para resolver  $V_1$  e assim por diante. Nós assumimos que o valor de  $b$  usado para resolver cada  $V_i$  durante o treinamento é guardado na memória e é usado para aprender  $\pi_n$ . O conjunto parcialmente ordenado  $V$  apenas aproxima  $L^{-1}(\pi_n)$  já que poderia existir outros conjuntos que iriam permitir que se aprendesse  $\pi_n$  mais eficientemente (ex: usando menores valores de  $b$  que permitissem LTS resolver as instâncias).

Na nossa implementação de *Bootstrap*, nós setamos  $b = 1$  e dobramos o valor se o algoritmo completar uma iteração sem resolver nenhuma das instâncias ainda não resolvidas. Nós também setamos  $b = 1$  toda vez que a política é atualizada. Trabalhos anteriores têm usado valores maiores de  $b$  que aumentam monotonicamente já que isso tende a reduzir esforço na aprendizagem de  $\pi_n$ . Nós usamos um  $b$  pequeno e o resetamos para 1 sempre que a política atualiza para produzir pequenos subconjuntos  $V_i$ . Para um  $b$  suficientemente grande nós teríamos que  $V_0 = D$  e o *Bootstrap* não iria prover nenhuma ordenação das instâncias. Ter subconjuntos menores  $V_i$  é importante para o problema da aprendizagem de currículo, pois todas as instâncias em  $V_i$  têm uma mesma posição na

**Algoritmo 1:** Bootstrap

---

**Entrada:**  $D, R, \pi_0$   
**Saída:**  $\pi, V$

```

1  $\pi \leftarrow \pi_0$   $d \leftarrow D$   $i \leftarrow 0$   $b \leftarrow 1$   $V \leftarrow []$ ;
2 enquanto  $V$  não contiver todas as instâncias de  $D$  faça
3    $v \leftarrow R(\pi, d, b)$ ;
4   se  $V$  contém ao menos uma instância resolvida então
5      $V[i] \leftarrow v$ ;
6      $i \leftarrow i + 1$ ;
7      $b \leftarrow 1$ ;
8     retire as instâncias solucionadas  $v$  de  $d$ 
9     treine a rede neural com o conjunto  $v$ , atualizando a política  $\pi$ 
10  senão
11     $b \leftarrow b \times 2$ 
12  fim
13 fim
14 retorne  $\pi, V$ 

```

---

ordenação parcial de  $V$ . Assim, quando apresentado para estudantes, as instâncias em  $V_i$  seriam apresentadas em uma ordem arbitrária. Por minimizar o tamanho de  $V_i$  nós minimizamos o número de instâncias que são apresentadas em ordem arbitrária aos estudantes. Para melhor visualizar o funcionamento do algoritmo de *Bootstrap*, apresentamos o pseudocódigo no Algoritmo 1.

Uma ordenação é dita total quando todas as instâncias presentes no conjunto têm uma posição única dentro da ordenação. Em contra ponto, uma ordenação é dita parcial quando existem instâncias que têm a mesma posição dentro de uma ordenação. Em nosso caso, uma ordenação parcial existe quando mais de uma instância faz parte de um mesmo conjunto de instâncias solucionadas juntas  $V_i$ , pois todas têm a mesma posição  $i$  dentro da ordenação.

## 6 Aprendizagem de Currículo com Bootstrap

Nós consideramos duas variações do problema de aprendizagem de currículo. Na primeira variação recebemos um pequeno conjunto de instâncias  $D_o$  e precisamos prover uma ordenação total para  $D_o$ . As instâncias são então apresentadas para os estudantes na ordenação definida pelo currículo. Chamamos essa variação de problema de ordenação; o ‘o’ subscrito em  $D_o$  significa ‘ordenação’.

Na segunda variação recebemos um grande conjunto  $D_s$  e um inteiro  $c$ ; precisamos então prover um subconjunto totalmente ordenado  $D'$  de  $D_s$  de tamanho  $c$ . O subconjunto  $D'$  é entregue como um currículo para estudantes. Chamamos essa variação de problema de ordenação e seleção; o ‘s’ subscrito em  $D_s$  significa ‘seleção’.

### 6.1 Problema de Ordenação

O número de instâncias no conjunto  $D_o$  entregue como entrada para o problema de ordenação é muito menor que o número de instâncias de treinamento normalmente requerido para treinar uma política neural (pessoas tendem a precisar de muito menos instâncias que redes neurais para aprender). Em vez de usar apenas  $D_o$  como o conjunto de entrada para o *Bootstrap*, nós entregamos como entrada um conjunto de treinamento maior onde  $D_o$  é um subconjunto. Então, rodamos o *Bootstrap* com este conjunto aumentado. Os índices  $i$  dos subconjuntos  $V_i$  determinam a ordem para as instâncias em  $D_o$ . Seja  $(s, g)$  e  $(s', g')$  duas instâncias em  $D_o$ . A instância  $(s, g)$  é considerada mais fácil que  $(s', g')$  se  $(s, g)$  está em  $V_i$  e  $(s', g')$  está em  $V_j$  e  $i < j$ .

Já que cada execução do *Bootstrap* pode resultar em uma política diferente devido à natureza aleatória da inicialização dos pesos da rede neural, nós rodamos o *Bootstrap* múltiplas vezes em vez de apenas uma. Nós determinamos a ordenação das instâncias usando o índice médio  $i$  dos subconjuntos  $V_i$  de cada instância de  $D_o$  através de todas as execuções rodadas. As várias execuções do sistema e nosso sistema de budget (ex:  $b = 1$ ) tornam improvável que duas instâncias em  $D_o$  vão sempre ter o mesmo índice médio, o que implicaria uma ordenação parcial. Em nossos experimentos, nosso sistema sempre retornou uma ordenação total para  $D_o$ .

### 6.2 Problema de Ordenação e Seleção

Assumimos que o tamanho de  $D_s$  é grande o suficiente para treinar uma política neural, então  $D_s$  é entregue ao *Bootstrap*. Similarmente ao problema de ordenação, nós também rodamos o *Bootstrap* múltiplas vezes e computamos o índice médio  $i$  do subcon-

junto  $V_i$  para cada instância em  $D_s$ , o que nos dá uma ordenação das instâncias em  $D_s$ . Nós então selecionamos  $c$  instâncias em  $D_s$  para formar um currículo como segue.

Seja  $\{o_1, o_2, \dots, o_m\}$  um conjunto totalmente ordenado de instâncias. Nós selecionamos instâncias com índice  $\lfloor \frac{i \times m}{(c-1)} \rfloor$  para  $i = 0, \dots, c-1$  para compor o currículo. Esse procedimento seleciona  $c$  instâncias equidistantes de acordo com seus índices. Por exemplo, se  $m = 1000$  e  $c = 5$ , nós selecionamos as instâncias com os índices 0, 250, 500, 750, 1000. Se o *Bootstrap* entrega uma ordenação parcial de  $D_s$ , nós ordenamos as instâncias com o mesmo rank arbitrariamente. Esse processo garante que o currículo tenha instâncias de dificuldades variadas, como avaliado pelo *Bootstrap*. Em particular, inclui a instância mais fácil e a mais difícil de  $D_s$ .

## 7 Desenho Experimental

Nós fizemos um estudo com usuários para avaliar a efetividade do currículo que o *Bootstrap* gerou para ambos os casos dos problemas de ordenação e o de ordenação e seleção. Nosso estudo foi feito de maneira online com voluntários advindos da plataforma Lab in the Wild (LitW) (REINECKE; GAJOS, 2015). Optamos por usar o LitW pois ele oferece uma gama mais diversa de participantes nos aspectos geográficos e demográficos do que outras plataformas. Em vez de receber uma compensação financeira, voluntários do LitW recebem informação sobre eles mesmos e sobre a conclusão do experimento como uma forma de compensação. Nosso estudo foi aprovado pelo comitê de ética e pesquisa da University of Alberta, Canadá.

Nós usamos um design entre-grupos onde cada participante recebeu um currículo aleatoriamente. Participantes inicialmente assinavam um termo de consentimento, respondiam questões demográficas e então tentavam resolver as instâncias do currículo antes de tentar resolver um conjunto de instâncias de teste. Participantes podiam pular instâncias ou deixar o experimento a qualquer momento. A próxima instância no currículo (ou no conjunto de teste) apenas era liberada caso o participante resolvesse a instância atual ou a pulasse. Após completar as instâncias de teste, os participantes poderiam retornar às instâncias puladas para tentar resolvê-las; porém não analisamos essas tentativas pós-estudo. A interface gráfica dos usuários do nosso sistema permitia que os participantes vissem as imagens dos *puzzles* que eles já haviam solucionado no experimento (imagens da interface do experimento podem ser encontradas a partir da Figura 6 do Apêndice). Participantes encerravam o experimento respondendo um questionários sobre o mesmo e recebendo um feedback sobre seus resultados.

### 7.1 Black and White Square puzzles

Nós usamos a categoria de *puzzles Black and White Square* (BWS) do jogo The Witness. No jogo original, o jogador aprende a resolver os *puzzles* BWS enquanto tenta solucioná-los dentro de um currículo cuidadosamente criado. A Figura 2 mostra 9 instâncias do currículo usado no jogo, onde o jogador resolve-as da esquerda para a direita. Inicialmente, apenas a instância mais a esquerda é acessível ao jogador e a próxima se torna acessível apenas quando o jogador soluciona a instância atual. Nos referimos a essas 9 instâncias como o currículo **witness-exato**. Em *puzzles* BWS, o jogador precisa desenhar uma linha começando do círculo localizado em uma borda externa do tabuleiro (observe a quina inferior-esquerda de todas as instâncias fora a primeira na Figura 2) e terminar em uma pequena linha que sai do tabuleiro. A linha desenhada deve particio-

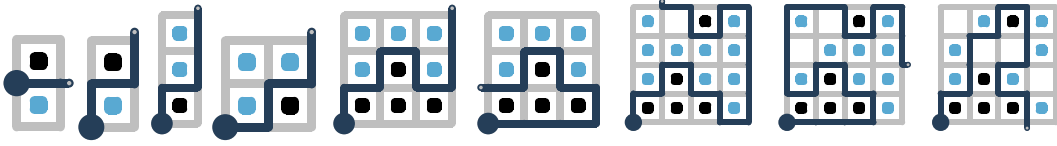


Figura 2 – Currículo original do jogo ‘The Witness.’ O jogador soluciona as instâncias da esquerda para a direita.

nar as células do tabuleiro de uma forma que quadrados de cores diferentes não estejam contidos na mesma partição. Se um jogador desenha uma linha do ponto de início até o ponto de fim que viola regra de separação de cores, a linha desenhada pisca em vermelho indicando que o jogador falhou em resolver o problema e o *puzzle* é reiniciado.

O currículo apresentado na Figura 2 permite que jogadores aprendam os conceitos necessários para resolver os *puzzles* em um processo de tentativa e erro. O jogo nunca instrui os jogadores sobre as condições para resolver os *puzzles* (e.g. sobre os pontos de início e fim, ou sobre separar quadrados de cores diferentes); o jogador precisa descobrir as condições enquanto tentam solucionar as instâncias no currículo.

Quando falamos de conceitos necessários alguns exemplos destes são: Aprender onde é o início e o fim de um caminho no tabuleiro; aprender que é necessário separar círculos de cores diferentes em áreas diferentes; aprender que pode ser necessário separar até mesmo círculos de cores iguais a depender do *puzzle*. Os currículos que almejamos criar devem ensinar esses e outros conceitos de forma que os alunos sejam capazes de absorvê-los sequencialmente enquanto resolvem as instâncias dos currículos.

## 7.2 Treinando o Modelo Neural

Utilizamos uma rede neural com 2 camadas convolucionais com 32 e 64 filtros de tamanho  $2 \times 2$  e unidades de ativação ReLU. As camadas convolucionais são ligadas a duas camadas totalmente conectadas com 2048 unidades ReLU. Finalmente, a última camada totalmente conectada se liga a uma camada Softmax com 4 saídas, uma para cada ação no jogo (desenhar o caminho de solução para uma das 4 direções cardinais). Depois de completar a  $j$ -th iteração do *Bootstrap*, treinamos a rede neural com o algoritmo Adam com passo de tamanho  $10^{-4}$  para minimizar a loss da entropia-cruzada do conjunto de treinamento dada por  $\cup_{i \leq j} V_i$ . Nós usamos mini-batches de tamanho 1024 e fazemos rodadas de treinamento até que o erro médio da entropia-cruzada de todo o conjunto de treinamento seja menor que 0.1.

Cada estado do jogo é representado por uma imagem de dimensões  $9 \times 9 \times 7$ . Os grids de tamanho  $9 \times 9$  são usados para representar *puzzles* tão grandes quanto  $4 \times 4$  (4 entradas para as células e 5 entradas para os caminhos que existem ao redor das células, por onde se desenha a linha de solução). Cada um dos 7 canais contém 0's ou 1's. Usamos

um canal para cada cor, com o valor 1 indicando a presença de um quadrado de uma dada cor, um canal para representar o ponto de entrada do caminho, uma para representar a ponta do caminho de resposta atual, um para representar o caminho desenhado até agora, um para denotar o ponto de destino do caminho, e um que aponta as células ativas do tabuleiro, ou seja as células que estão sendo usadas de alguma forma naquele problema (nós usamos a mesma representação para *puzzles* menores e as células extras que não são utilizadas são marcadas nesse canal).

Nós desenvolvemos um gerador de *puzzles* BWS. Nosso gerador começa com um tabuleiro vazio com a posição de início no canto inferior-esquerdo (como a maioria dos *puzzles* no jogo). Então, ele coloca a posição de fim em uma borda aleatória do tabuleiro que seja diferente da posição inicial. Ele então gera um caminho aleatório entre a posição de início e a de fim, o que particiona as células em regiões desconexas. Para cada partição nós escolhemos uma das duas cores aleatoriamente com probabilidades iguais. Então, para cada célula em uma partição, colocamos um quadrado da cor escolhida com 80% de chance, ou a deixamos vazia. Nós geramos um total de 2360 *puzzles*: 10 com tabuleiros de tamanho  $1 \times 2$ ; 50 com tamanho  $1 \times 3$ ; 300 com tamanho  $2 \times 2$ ; 1000 com tamanho  $3 \times 3$ ; 1000 com tamanho  $4 \times 4$ . Apenas tendo três opções de preenchimento por quadrado (branco, preto ou sem cor) e olhando apenas para os tabuleiros de tamanho  $4 \times 4$  o gerador é capaz de criar pelo menos 7 milhões de possíveis instâncias, ou seja, o espaço de possíveis *puzzles* para este problema é vasto.

Nós usamos as 9 instâncias do currículo witness-exato como o conjunto  $D_o$  para o problema de ordenação. A união das 2360 instâncias geradas e as 9 instâncias do witness-exato formam tanto o conjunto aumentado para o problema de ordenação quanto o conjunto do problema de ordenação e seleção.

Nós rodamos o *Bootstrap* com a LTS 100 vezes para computar o ranking médio dos índices das instâncias e ordená-las. O *Bootstrap* obteve exatamente a mesma ordem usada no jogo original para o problema de ordenação com as 9 instâncias do witness-exato. Partindo do pressuposto que o designer do jogo criou currículo quase ótimo para estudantes humanos, esse resultado é evidência que o *Bootstrap* com LTS pode ser um efetivo modelo para resolver o problema de ordenação da geração de currículos. Khan, Mutlu e Zhu (2011) também pressupõem que pessoas criam currículos quase ótimos e performaram um experimento similar, onde eles compararam currículos criados por humanos a currículos gerados por máquinas.

### 7.3 Currículos para o Estudo

Nós utilizamos o currículo **witness-exato** para representar tanto o currículo do jogo original quanto a solução do *Bootstrap* para o problema de ordenação. Nós também

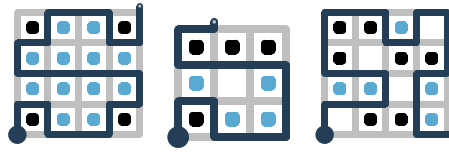


Figura 3 – As instâncias de teste usadas em nosso estudo. A primeira instância é advinda do jogo e é fixada para todos os participantes. As outras duas são apenas representativas das instâncias de teste.

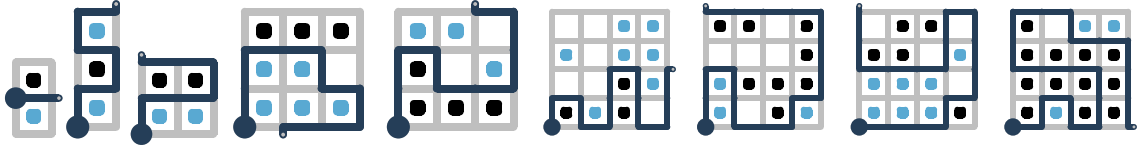


Figura 4 – A solução de nosso sistema para o problema da ordenação e seleção (currículo equidistante).

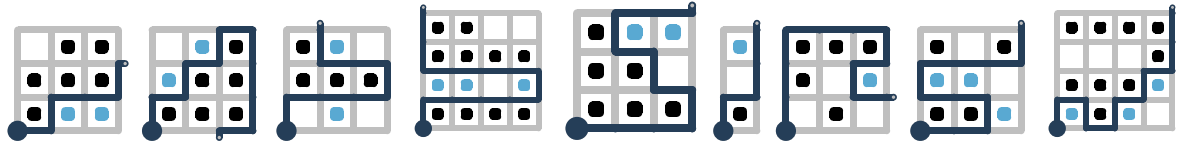


Figura 5 – Exemplo de currículo aleatório.

avaliamos duas baselines que representam diferentes ordenações destes mesmos 9 *puzzles*: Um apresenta-os em ordem reversa (**witness-reverso**) e o outro apresenta as instâncias em ordem aleatória (**witness-aleatório**). Para o **witness-aleatório**, as 9 instâncias eram misturadas independentemente para cada participante que tivesse sido sorteado para este currículo. Essas baselines permitiram que nós avaliássemos se a ordenação escolhida pelo designer do jogo e encontrada pelo *Bootstrap* é superior a outras alternativas mais simples.

Também avaliamos a solução do *Bootstrap* para o problema de ordenação e seleção, que é apresentada na Figura 4. Nós usamos  $c = 9$  para permitir uma comparação justa entre os currículos baseados nas instâncias do jogo original. Nós chamamos esse currículo de **equidistante** pois ele seleciona instâncias equidistantes de acordo com a ordenação retornada pelo *Bootstrap*. Finalmente, nós também avaliamos o desempenho de um currículo composto pela seleção aleatória de 9 instâncias dentre os 2369 *puzzles* usados em nossos experimentos. Chamamos esse currículo de **aleatório**. As 9 instâncias são selecionadas aleatoriamente e independentemente para cada participante que foi sorteado para solucionar este currículo. A Figura 5 mostra um exemplo de currículo aleatório.

## 7.4 Instâncias de Teste para o Estudo

A Figura 3 mostra um conjunto representativo de instâncias de teste usadas em nosso estudo. A primeira instância no conjunto de teste (o *puzzle* mais a esquerda na

figura), a qual chamamos de p10 por ser o 10<sup>o</sup> *puzzle* a ser apresentado aos participantes, foi idêntica para todos os participantes. A instância p10 também foi retirada do jogo The Witness. No jogo The Witness, após resolver as 9 instâncias da Figura 2, o jogador resolve outros tipos de *puzzles* antes de encontrar p10. Nós usamos p10 em nosso conjunto de teste pois, no jogo original, o jogador é capaz de destravar uma porta após resolvê-lo, como se o jogo usasse p10 para testar o conhecimento que o jogador adquiriu após resolver os 9 *puzzles* iniciais. Os outros dois *puzzles* mostrados na figura podem variar para cada participante. Porém, nós controlamos o tamanho de caminhos de solução para manter um nível de igualdade (12 e 20 ações para a segunda e a terceira instâncias de teste respectivamente).

## 7.5 Métricas

Nós avaliamos os currículos descritos usando proxies que indicam como o currículo se relacionou com a persistência dos estudantes, esforço e performance. Para a persistência, nós usamos as taxas de desistência dos participantes (ex: retenção do usuário), onde um participante é considerado desistente caso ele saia do experimento antes de tentar ao menos a primeira instância de teste. Nós também contabilizamos o número de *puzzles* que cada participante resolveu antes de desistir. Para a questão do esforço, contamos o número de vezes que o participante tentou resolver cada instância de treinamento. Uma tentativa é contada quando o caminho é resetado e o participante pode começar a resolver do início novamente. Este resete acontece quando o participante intencionalmente aperta o botão de ‘resete’ ou quando conecta o início ao fim do caminho sem separar corretamente quadrados de cores diferentes. Para a performance, reportamos o tempo requerido e o número de *puzzles* resolvidos por participante. Também reportamos o número de passos reversos (*backtracks*) cada participante performou. Um passo de backtrack é contado quando alguém desfaz um segmento do caminho no tabuleiro, revertendo um movimento feito anteriormente. Para todas as métricas nós reportamos sua média (M) e seu desvio padrão (DP).

## 7.6 Testes Estatísticos

Já que os dados são binomiais, para os resultados sobre desistência nós usamos o teste Chi-Squared e teste pareado de Fisher para comparar as duas médias. Para os outros testes usamos os não-paramétricos Kruskal-Wallis para todos os currículos e Mann-Whitney para comparar os pares de médias. Em ambos os casos, nossa hipótese nula ( $h_0$ ) é de que não existe diferença entre as médias de cada currículo para os dados testes que serão abordados. Falamos que a diferença é estatisticamente significativa se o  $p$ -valor dos testes for menor que .05, neste caso rejeitamos a hipótese  $h_0$ . Usamos letras sobrescritas para

denotar diferenças estatisticamente significantes. Se dois números na mesma coluna têm diferentes letras sobrescritas, então a diferença entre eles é estatisticamente significativa. Por exemplo, na Tabela 1, a diferença na taxa de desistência entre witness-exato e witness-reverso é estatisticamente significativa, enquanto a diferença entre equidistante e witness-exato não é. Nós não usamos letras sobrescritas se o  $p$ -valor do teste Chi-Squared ou Kruskal-Wallis entre todas as médias for maior que .05.

## 8 Resultados do Estudo com Usuários

Neste capítulo reportamos os dados coletados entre Dezembro de 2018 e Agosto de 2021 através da plataforma LitW. Nosso experimento teve um total de 689 participantes dos quais 82 já haviam jogado The Witness anteriormente, resultando em um total de 607 participantes válidos. A idade média dos participantes foi de 28.8 e o desvio padrão foi de 11.09; a idade mínima foi de 18 e a máxima de 78; 238 participantes se identificaram como do sexo feminino, 205 como do masculino, 8 como outro e 156 não o especificaram. 499 já receberam ou estavam concluindo seu ensino superior; 108 já haviam completado educação primária e/ou ensino médio. Também perguntamos duas questões sobre as experiências dos participantes com jogos e, mais especificamente, com resolução de *puzzles*. Usamos uma escala do tipo 7-Likert de 1, ‘nunca jogo jogos/resolvo *puzzles*’ até 7 ‘sempre jogo jogos/resolvo *puzzles*’. A mediana das respostas para jogar foi de 5 (IQR = 2) e para resolver *puzzles* foi de 4 (IQR = 2). A maioria dos participantes reportou que joga jogos e resolve *puzzles* ocasionalmente. A distribuição demográfica é similar através de todos os currículos.

### 8.1 Retenção de Usuário

Uma maior taxa de desistência pode ocorrer devido ao currículo falhar em balancear a dificuldade de suas instâncias. A coluna do meio da Tabela 1 mostra a taxa de desistência de cada currículo. Por exemplo, 39% dos participantes que receberam o currículo witness-exato deixaram o experimento antes de tentar as instâncias de teste. A última coluna na Tabela 1 mostra a média e o desvio padrão do número de *puzzles* resolvidos por cada participante antes de desistir. Por exemplo, dos 9 *puzzles* no currículo, participantes do witness-exato resolveram 3.31 *puzzles* em média antes de desistir com desvio padrão de 1.99.

O currículo com a melhor retenção de usuário é o witness-exato (39% de desistência) enquanto os dois piores são o aleatório (65%) e o witness-reverso (64%). Participantes também resolveram mais *puzzles* antes de desistir do witness-exato. Esses resultados sugerem que a ordem importa e que a ordem escolhida pelo nosso sistema encoraja maiores taxas de retenção de usuário quando comparada com as baselines de witness-reverso e witness-aleatório.

Ao passo de que a retenção não foi significativamente diferente entre os currículos equidistante e witness-exato, participantes resolveram mais *puzzles* antes de desistir quando foi dado a eles o currículo gerado pelo problema da ordenação e seleção (equidistante) em comparação com as outras baselines. Além disso, a taxa de desistência do

Currículo Selecionado	Desistência	# Resolvido Antes de Desistir
witness-exato	39% <sup>a</sup>	3.31 <sup>a</sup> (1.99)
witness-aleatório	50% <sup>a</sup>	1.29 <sup>b</sup> (1.87)
witness-reverso	64% <sup>b</sup>	0.42 <sup>c</sup> (1.39)
aleatório	65% <sup>b</sup>	0.70 <sup>d</sup> (1.65)
equidistante	48% <sup>a</sup>	2.46 <sup>e</sup> (1.98)

Tabela 1 – A taxa de desistência e o número de *puzzles* resolvidos, como M (DP), por participante antes de desistir. Letras sobrescritas (*a, b, c, d, e*) significam diferentes distribuições no teste pareado de Fisher e Mann-Whitney para a primeira e segunda coluna respectivamente.

Currículo Selecionado	Número de <i>puzzles</i> Resolvidos		
	Currículo	Teste	Primeiro Teste
witness-exato	8.04 <sup>a</sup> (1.45)	2.05 (0.85)	0.35 (0.48)
witness-aleatório	7.55 <sup>a</sup> (1.96)	1.87 (0.96)	0.32 (0.47)
witness-reverso	6.76 <sup>b</sup> (2.13)	1.74 (1.00)	0.25 (0.43)
aleatório	7.34 (2.54)	2.13 (1.01)	0.45 (0.50)
equidistante	7.29 (1.75)	2.00 (0.96)	0.39 (0.49)

Tabela 2 – Número de *puzzles* solucionados como M (DP). Letras sobrescritas (*a, b*) significam diferentes distribuições no teste pareado de Mann-Whitney, usado apenas entre os três currículos do witness.

currículo equidistante foi significativamente menor que as do witness-reverso e aleatório. Esses resultados sugerem que o currículo gerado com nosso sistema retém participantes por mais tempo.

## 8.2 Número de Instâncias Resolvidas

Como estamos interessados em analisar os resultados das instâncias de teste, consideramos apenas os dados daqueles que tentaram solucionar todos os *puzzles* de seus respectivos currículos e pelo menos uma instância de teste. Houveram 282 participantes não-desistentes (112 do sexo feminino, 105 do masculino, 4 de outro e 61 que não especificaram). A média de suas idades foi de 29.3 (DP = 10.4). As distribuições de frequência em que participantes jogam jogos e resolvem *puzzles*, e ainda o nível de educação dos participantes foi similar ao reportado nas médias gerais dos participantes.

A Tabela 2 mostra quantas instâncias dos currículos (o máximo é 9) e dos testes (o máximo é 3) em média e DP os participantes resolveram. Ela também mostra o número médio de *puzzles* p10 (Primeiro Teste) resolvidos por participante (o máximo é 1). Podemos ver que foram resolvidas significativamente menos instâncias no currículo witness-reverso em comparação com o que foi resolvido no witness-exato ou witness-aleatório. Esse resultado reforça aqueles que tivemos baseados na retenção dos participantes: a ordem dos *puzzles* influencia no número de *puzzles* resolvidos até mesmo para aqueles que com-

Currículo	Tentativas		
	Selecionado	Currículo	Primeiro Teste
witness-exato	2.88 (1.59)	4.22 <sup>a</sup> (3.94)	7.96 (7.79)
witness-aleatório	3.90 (1.96)	3.85 <sup>a</sup> (3.57)	6.73 (6.29)
witness-reverso	3.72 (3.69)	4.31 <sup>a</sup> (3.75)	7.96 (6.18)
aleatório	4.02 (2.08)	3.45 <sup>a,b</sup> (3.01)	7.52 (7.15)
equidistante	3.44 (1.93)	2.44 <sup>b</sup> (2.28)	5.23 (5.57)

Tabela 3 – Número de tentativas, como M (DP), para as instâncias do currículo, de teste e para a primeira instância de teste (p10). Letras sobrescritas (*a*, *b*) significam diferentes distribuições no teste pareado Mann-Whitney.

pletaram o currículo. Apenas comparamos os pares entre witness-exato, witness-aleatório e witness-reverso pois esses currículos usam o mesmo conjunto de instâncias e diferem apenas em sua ordenação.

Nenhuma diferença estatística foi encontrada quando analisando as instâncias de teste e o primeiro *puzzle* de teste, apesar de aparentar que os usuários resolveram mais instâncias e tiveram mais sucesso no currículo aleatório. Nós suspeitamos que a grande média de *puzzles* resolvidos em aleatório pode estar relacionada com o efeito de um viés de sobrevivência (WALD, 1980; LOCKWOOD, 2021). O currículo aleatório tem a maior taxa de desistência entre todos os currículos (veja na Tabela 1); pode ter ocorrido de que apenas os mais habilidosos participantes em solucionar *puzzles* tenham alcançado as instâncias de teste deste currículo.

## 8.3 Esforço dos Usuários

Agora observaremos as métricas de esforço de usuário. O resultado que reportamos aqui é apenas para os participantes não-desistentes, não importando se eles resolveram ou não a dada instância.

### 8.3.1 Tentativas

A Tabela 3 apresenta o número de tentativas dos *puzzles* no currículo, nos *puzzles* de teste e apenas no primeiro *puzzle* de teste. Ao passo de que os resultados das instâncias dos currículos não são estatisticamente significantes, eles sugerem que o witness-exato requer menos esforço dos participantes, já que os números médios de tentativas é apenas 2.88 e os participantes do witness-exato resolvem um número maior de instâncias do currículo (veja a coluna 2 na Tabela 2). Existe significância estatística na diferença entre equidistante e todos os currículos fora aleatório nas instâncias de teste. Enquanto o número médio de instâncias de teste resolvidas parece ser similar para todos os currículos (veja a coluna de instâncias de Teste na Tabela 2), os participantes do currículo equidistante

Currículo	backtracks		
	Selecionado	Currículo	Primeiro Teste
witness-exato	0.84 <sup>a</sup> (0.98)	5.14 (6.18)	10.55 (15.36)
witness-aleatório	1.21 <sup>a</sup> (1.41)	4.99 (5.30)	10.57 (13.78)
witness-reverso	0.92 <sup>a</sup> (0.89)	3.79 (4.11)	8.64 (11.41)
aleatório	1.41 <sup>a</sup> (2.27)	5.05 (5.44)	11.20 (13.36)
equidistante	2.30 <sup>b</sup> (2.39)	3.78 (3.99)	7.67 (9.80)

Tabela 4 – Número de *backtracks*, como M (DP), por tipo de instância e currículo. Letras sobrescritas (*a*, *b*) significam diferentes distribuições no teste pareado Mann-Whitney.

Currículo	Tempo em Segundos		
	Selecionado	Currículo	Primeiro Teste
witness-exato	189.6 (134.97)	206.1 (165.88)	116.7 (91.92)
witness-aleatório	166.1 (88.92)	134.6 (91.94)	86.6 (88.42)
witness-reverso	162.1 (72.25)	169.4 (53.86)	104.9 (73.97)
aleatório	173.9 (95.53)	129.1 (79.51)	76.7 (56.51)
equidistante	383.0 (265.59)	106.7 (70.67)	57.6 (36.44)

Tabela 5 – Tempo, como M (DP), por tipo de instância e currículo.

resolvem os problemas com menos tentativas. Um padrão similar aparenta estar emergindo para o caso de observarmos apenas a primeira instância de teste.

### 8.3.2 Backtracks

O número médio de *backtracks* por *puzzle* dos participantes de cada currículo é mostrado na Tabela 4 (ex: cada participante performou uma média de 0.84 *backtracks* por *puzzle* no currículo witness-exato). Participantes do witness-exato fizeram a menor quantidade de *backtracks* para as instâncias do currículo enquanto os do equidistante fizeram significativamente mais *backtracks* que todos os outros currículos. Esse resultado sugere que o currículo equidistante requer mais esforço dos participantes. Interessantemente, apesar de requerer mais esforço, equidistante tem uma taxa de desistência significativamente menor que witness-reverso e aleatório (veja a Tabela 1). Nós conjecturamos que equidistante impõe a quantidade certa de desafio para os estudantes, onde eles devem se esforçar para resolver os *puzzles*, tornando-se uma forma de conquista ou prêmio (TSOUROUNIS; EPP, 2016). Esse tipo de investimento de esforço já mostrou ser capaz de encorajar atividades continuadas (FESTINGER, 1961). witness-reverso parece ser muito difícil, o que pode desencorajar pessoas. Apesar de não ser estatisticamente significante, esse comportamento parece ser revertido nas instâncias de teste, onde os participantes do equidistante performaram menos *backtracks*, sugerindo que o currículo os preparou bem para as instâncias de teste.

### 8.3.3 Tempo

O tempo médio que os participantes gastaram nas instâncias do currículo, nas de teste e na primeira instância de teste é apresentado na Tabela 5. Os resultados de tempo sugerem um padrão similar ao observado nos *backtracks*: participantes do currículo equidistante parecem gastar mais tempo resolvendo as instâncias do currículo e menos resolvendo as de teste.

## 8.4 Discussão

Os resultados das métricas de esforço do usuário sugerem que o currículo equidistante requer mais esforço dos mesmos. Nós reconhecemos que o esforço requerido é bem balanceado com o conhecimento dos participantes e sua velocidade de aprendizado já que a taxa de desistência do equidistante é significativamente menor que os currículos witness-reverso e aleatório. equidistante também parece preparar melhor os estudantes para resolver as instâncias de teste, já que eles precisaram de menos tentativas, menos *backtracks* e menos tempo para resolver o mesmo número de instâncias de teste em comparação com os outros currículos.

A última instância do currículo equidistante, a qual nos referimos como p9, oferece possíveis explicações para os níveis de esforço dos usuários deste currículo. A instância p9 requer que o caminho de solução separe quadrados da mesma cor (veja a instância mais a direita na Figura 4), o que é uma característica rara que torna o *puzzle* difícil tanto para o sistema quanto para as pessoas. Chamaremos essa característica de ‘separação da mesma cor’. Lembre-se que p9 é a instância mais difícil de acordo com a ordenação do nosso sistema e, ao mesmo tempo que o número médio de tentativas para todo o currículo equidistante é 3.44, p9 sozinho tem uma média de tentativas maior que 12 (resultados apresentados na Tabela 11 do material suplementar). Nenhuma outra instância requer mais tentativas, *backtracks* e tempo que a p9 do equidistante. Apesar de conter a instância mais complexa do estudo, o currículo equidistante tem uma taxa de desistência significativamente menor que witness-reverso e aleatório, e comparativamente tão baixa quanto witness-exato (o currículo criado pelo designer profissional do jogo e encontrado por nosso sistema como resultado do problema de ordenação). Nós suspeitamos que isso ocorre pois os estudantes estão motivados por terem conseguido resolver os *puzzles* anteriores ao p9 e terem aprendido boa parte dos conceitos requeridos para resolver *puzzles* do tipo BWS.

Como p9, o *puzzle* de teste p10 também tem a característica de separação da mesma cor (veja a Figura 3). Talvez o designer do jogo tenha guardado um *puzzle* com esta característica como um teste para o jogador. Interessantemente, nosso sistema seleciona uma instância com essa rara característica para ser parte de seu currículo. Um bom

currículo deveria cobrir todos os aspectos importantes que o estudante deve conhecer e o currículo equidistante é o único que inclui uma instância com essa característica. É possível argumentar que p10 é parte do currículo encontrado no jogo original; o jogo apenas faz com que os jogadores resolvam outros tipos de *puzzles* depois de terminar as instâncias do currículo (wintess-exact) antes de resolver p10. Neste caso, o designer do jogo escolheu p10 para ser a última instância do currículo, o que é similar com a escolha do nosso sistema que fez a escolha de p9 para ser a última do currículo equidistante.

As análises de número de *puzzles* resolvidos e esforço de usuário são provavelmente influenciadas pelo viés de sobrevivência que favorece witness-reverso, o qual teve uma maior taxa de desistência. Para um currículo que começa com instâncias tão difíceis, apenas aqueles que são mais habilidosos em resolver *puzzles* “sobrevivem” até as instâncias de teste. Esse viés sobre os que permaneceram é provavelmente o porquê do witness-reverso requerir menos *backtracks* que o witness-exato quando os participantes estão resolvendo as instâncias de teste. Algumas de nossas conclusões não são afetadas por este viés. Por exemplo, o currículo equidistante tem uma taxa de desistência significativamente menor que witness-reverso e números significativamente menores de tentativas nas instâncias de teste. As conclusões entorno da taxa de desistência não são afetadas pelo viés de sobrevivência. Em vez disso, elas provêm evidência para a existência desse viés amostral.

## 9 Conclusão

Neste trabalho, avaliamos o uso de *Bootstrap* e LTS para aprender um currículo para humanos em uma tarefa de decisão sequencial. Nossa avaliação comparou dois currículos gerados com outras baselines. O primeiro currículo gerado pegou um conjunto de instâncias e as ordenou de uma forma que fosse favorecer a aprendizagem se as instâncias fossem resolvidas em ordem. O segundo pegou um conjunto maior de instâncias e retornou um subconjunto ordenado para servir como um currículo. Usamos LTS e *Bootstrap* para resolver ambos os problemas no contexto de *puzzles* do jogo The Witness. Para o primeiro currículo, a solução de nosso sistema encontrou uma ordenação que se igualou exatamente a de *puzzles* encontrados no jogo original, o que por si só é evidência que nosso sistema foi capaz de produzir um bom currículo. Nós avaliamos os currículos que nosso sistema gerou e descobrimos que eles se comparam favoravelmente com outras baselines em termos de retenção e esforço de usuário quando resolvendo um conjunto de instâncias de teste.

Este trabalho trás um primeiro passo na direção de um sistema que pode ser capaz de selecionar e gerar currículos de diversos assuntos, dada uma base de dados do determinado assunto. Um sistema como esse será capaz de reduzir a jornada de trabalho de um profissional do ensino ou direcioná-la mais a outras atividades, dado que boa parte do tempo gasto por esses profissionais se dá na preparação do material didático. Para a inteligência artificial educacional, nosso trabalho trás mais uma maneira de atacar o problema que os ITSs vêm trabalhando à décadas, porém com muito menos esforço humano envolvido dado o fato do modelo aprender e selecionar as instâncias de currículo sozinho e a possibilidade de que a base de dados que o sistema usará também pode ser gerada automaticamente como fizemos com nosso gerador de *puzzles* BWS.

# Referências

- ARFAEE, S. J.; ZILLES, S.; HOLTE, R. C. Learning heuristic functions for large state spaces. **Artificial Intelligence**, v. 175, n. 16, p. 2075–2098, 2011.
- BENGIO, Y. et al. Curriculum learning. In: **Proceedings of the 26th Annual International Conference on Machine Learning**. [S.l.]: Association for Computing Machinery, 2009. p. 41–48.
- BOSCH, N. et al. Automatic Detection of Learning-Centered Affective States in the Wild. In: **IUI 2015**. Atlanta, Georgia, USA: ACM Press, 2015. p. 379–388. Disponível em: <<http://dl.acm.org/citation.cfm?id=2701397>>.
- CAKMAK, M.; LOPES, M. Algorithmic and human teaching of sequential decision tasks. In: **Proceedings of the AAI Conference on Artificial Intelligence**. [S.l.]: AAAI Press, 2012. p. 1536–1542.
- CHI, M. et al. Empirically evaluating the application of reinforcement learning to the induction of effective and adaptive pedagogical strategies. **User Modeling and User-Adapted Interaction**, v. 21, n. 1, p. 137–180, 2011.
- CORBETT, A. et al. The impact on learning of generating vs. selecting descriptions in analyzing algebra example solutions. In: **Proceedings of the 7th international conference on Learning sciences**. International Society of the Learning Sciences, 2006. (ICLS '06), p. 99–105. ISBN 0-8058-6174-2. Disponível em: <<http://dl.acm.org/citation.cfm?id=1150034.1150049>>.
- DOROUDI, S.; ALEVEN, V.; BRUNSKILL, E. Where's the Reward? A Review of Reinforcement Learning for Instructional Sequencing. **International Journal of Artificial Intelligence in Education**, v. 29, n. 4, p. 568–620, dez. 2019. ISSN 1560-4306. Disponível em: <<https://doi.org/10.1007/s40593-019-00187-x>>.
- FENZA, G.; ORCIUOLI, F.; SAMPSON, D. Building Adaptive Tutoring Model Using Artificial Neural Networks and Reinforcement Learning. In: **2017 IEEE 17th International Conference on Advanced Learning Technologies (ICALT)**. Timisoara, Romania: IEEE, 2017. p. 460–462.
- FESTINGER, L. The psychological effects of insufficient rewards. **American Psychologist**, v. 16, n. 1, p. 1–11, 1961. ISSN 1935-990X(Electronic);0003-066X(Print).
- GOLDMAN, S.; KEARNS, M. On the complexity of teaching. **Journal of Computer and System Sciences**, Academic Press, Inc., v. 50, n. 1, p. 20–31, 1995.
- GRAESSER, A. C.; D'MELLO, S. Chapter Five - Emotions During the Learning of Difficult Material. In: Brian H. Ross (Ed.). **Psychology of Learning and Motivation**. Academic Press, 2012, (The Psychology of Learning and Motivation, Volume 57). p. 183–225. ISBN 0079-7421. Disponível em: <<http://www.sciencedirect.com/science/article/pii/B9780123942937000054>>.

- HART, P. E.; NILSSON, N. J.; RAPHAEL, B. A formal basis for the heuristic determination of minimum cost paths. **IEEE transactions on Systems Science and Cybernetics**, IEEE, v. 4, n. 2, p. 100–107, 1968.
- KHAN, F.; MUTLU, B.; ZHU, J. How do humans teach: On curriculum learning and teaching dimension. In: **Advances in Neural Information Processing Systems**. [S.l.: s.n.], 2011. v. 24, p. 1–9.
- KOBAYASHI, H.; SHINOHARA, A. Complexity of teaching by a restricted number of examples. In: **Conference on Learning Theory**. [S.l.: s.n.], 2009.
- LINDSEY, R. V. et al. Optimizing instructional policies. In: BURGESS, C. J. C. et al. (Ed.). **Advances in Neural Information Processing Systems**. [S.l.: s.n.], 2013. v. 26.
- LOCKWOOD, D. **Foiled by the winners: how survivor bias deceives us**. S.l.: Greenleaf Book Group LLC, 2021. OCLC: 1257479685. ISBN 978-1-62634-880-6.
- MATSUDA, N. et al. Evaluating a Simulated Student Using Real Students Data for Training and Testing. In: **User Modeling 2007**. Springer, Berlin, Heidelberg, 2007. (Lecture Notes in Computer Science), p. 107–116. ISBN 978-3-540-73077-4 978-3-540-73078-1. Disponível em: <[https://link.springer.com/chapter/10.1007/978-3-540-73078-1\\_14](https://link.springer.com/chapter/10.1007/978-3-540-73078-1_14)>.
- ORSEAU, L. et al. Single-agent policy tree search with guarantees. In: **Advances in Neural Information Processing Systems 31**. [S.l.]: Curran Associates, Inc., 2018. p. 3201–3211.
- ORSEAU, L.; LELIS, L. H. S. Policy-guided heuristic search with guarantees. In: **AAAI Conference on Artificial Intelligence**. AAAI Press, 2021. p. 12382–12390. Disponível em: <<https://ojs.aaai.org/index.php/AAAI/article/view/17469>>.
- PARDOS, Z. A.; DADU, A. dAFM: Fusing Psychometric and Connectionist Modeling for Q-matrix Refinement. **Journal of Educational Data Mining**, v. 10, n. 2, p. 1–27, out. 2018. ISSN 2157-2100. Number: 2. Disponível em: <<https://jedm.educationaldatamining.org>>.
- PATIL, K. R. et al. Optimal teaching for limited-capacity human learners. In: GHAHRAMANI, Z. et al. (Ed.). **Advances in Neural Information Processing Systems**. [S.l.: s.n.], 2014. v. 27.
- RAFFERTY, A. N. et al. Faster teaching by pomdp planning. In: BISWAS, G. et al. (Ed.). **Artificial Intelligence in Education**. [S.l.]: Springer Berlin Heidelberg, 2011. p. 280–287.
- REINECKE, K.; GAJOS, K. Z. Labyrinthwild: Conducting large-scale online experiments with uncompensated samples. In: . [S.l.]: Association for Computing Machinery, 2015. p. 1364–1378.
- RUNZRAT, S.; HARFIELD, A.; CHAROENSIRIWATH, S. Applying Item Response Theory in Adaptive Tutoring Systems for Thai Language Learners. In: **2019 11th International Conference on Knowledge and Smart Technology (KST)**. Phuket, Thailand: IEEE, 2019. p. 67–71. ISSN: 2374-314X.

- SILVER, D. et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. **Science**, American Association for the Advancement of Science, v. 362, n. 6419, p. 1140–1144, 2018.
- TAYLOR, M. E. Assisting transfer-enabled machine learning algorithms: Leveraging human knowledge for curriculum design. In: **The AAAI Spring Symposium**. [S.l.]: AAAI, 2009. p. 141–143.
- TSOUROUNIS, S.; EPP, C. D. Learning dashboards and gamification in MALL: Design guidelines in practice. In: PALALAS, A.; ALLY, M. (Ed.). **The International Handbook of Mobile-Assisted Language Learning**. Beijing, China: China Central Radio & TV University Press Co., Ltd., 2016. p. 370–398. ISBN 978-7-304-07464-7.
- VANLEHN, K. The Behavior of Tutoring Systems. **Int. J. Artif. Intell. Ed.**, v. 16, n. 3, p. 227–265, ago. 2006. ISSN 1560-4292. Disponível em: <<http://dl.acm.org/citation.cfm?id=1435351.1435353>>.
- WALD, A. **A Reprint of 'A Method of Estimating Plane Vulnerability Based on Damage of Survivors**. [S.l.], 1980. Section: Technical Reports. Disponível em: <<https://apps.dtic.mil/sti/citations/ADA091073>>.
- WHITEHILL, J. Optimal teaching machines. In: . [S.l.: s.n.], 2009.
- ZHU, X. Machine teaching: An inverse problem to machine learning and an approach toward optimal education. In: **Proceedings of the AAAI Conference on Artificial Intelligence**. [S.l.: s.n.], 2015. v. 29, n. 1.
- ZHU, X.; LIU, J.; LOPES, M. No learner left behind: On the complexity of teaching multiple learners simultaneously. In: **Proceedings of the International Joint Conference on Artificial Intelligence**. [S.l.: s.n.], 2017. p. 3588–3594.
- ZHU, X. et al. **An Overview of Machine Teaching**. 2018.
- ZILLES, S. et al. Models of cooperative teaching and learning. **Journal of Machine Learning Research**, v. 12, n. 11, p. 349–384, 2011.

# Apêndices

# APÊNDICE A – Material Suplementar

## A.1 Retenção de Usuário

A Tabela 6 apresenta os  $p$ -valores para as comparações par-a-par das taxas de desistência. O teste de contingência de Chi-Square para a taxa de desistência de todos os currículos resultou em um valor de Chi-Square de 25.16,  $p$ -valor de  $4.65 \times 10^{-5}$ , e effect size ( $d$ ) de 0.41.

A Tabela 7 apresenta os  $p$ -valores para o número de problemas que os participantes conseguiram resolver antes de desistir do experimento. O teste de Kruskal-Wallis entre todos os currículos resultou em um valor de 133.13,  $p$ -valor de  $8.32 \times 10^{-28}$  e effect size ( $d$ ) de 1.04.

## A.2 Número de Instâncias Resolvidas

A Tabela 8 apresenta os  $p$ -valores para o número de instâncias dos currículos que os participantes não-desistentes de witness-exato, witness-reverso e witness-aleatório resolveram. Um teste de Kruskal-Wallis entre os 3 currículos resultou em um valor de 11.74,  $p$ -valor de  $2.81 \times 10^{-3}$ , e um effect size ( $d$ ) de 0.48.

O teste de Kruskal-Wallis para o número de instâncias de teste que os participantes não-desistentes resolveram resultou num valor de 6.21,  $p$ -valor de 0.18, e effect size ( $d$ ) de 0.17. O teste de Kruskal-Wallis para a primeira instância de teste (p10) entre os participantes não-desistentes resultou em 4.74,  $p$ -valor de 0.31, e effect size ( $d$ ) de 0.10. Como estes valores não são estatisticamente significantes, nós não fizemos testes pareados. A Tabela 9 apresenta a porcentagem de participantes não-desistentes que resolveram cada puzzle para cada currículo.

## A.3 Esforço dos Usuários

O teste de Kruskal-Wallis para o número de tentativas nas instâncias do currículo resultou em um valor de 1.22,  $p$ -valor de 0.87, e effect size ( $d$ ) de 0.20. O mesmo teste para o número de tentativas nas instâncias de teste resultou em uma estatística de 9.93,  $p$ -valor de 0.04, e effect size ( $d$ ) de 0.29. A Tabela 10 apresenta as comparações pareadas entre os currículos nas instâncias de teste. O teste de Kruskal-Wallis para o número de tentativas na primeira instância de teste resultou em um valor de 7.34,  $p$ -valor de 0.11, e effect size ( $d$ ) de 0.22. A Tabela 11 apresenta o número médio de tentativas por puzzle e

Par de Currículos	$p$
witness-exato & witness-aleatório	$1.15 \times 10^{-1}$
witness-exato & witness-reverso	$7.18 \times 10^{-5}$
witness-exato & aleatório	$4.68 \times 10^{-5}$
witness-exato & equidistante	$1.87 \times 10^{-1}$
equidistante & witness-aleatório	$7.89 \times 10^{-1}$
equidistante & witness-reverso	$1.42 \times 10^{-2}$
equidistante & aleatório	$8.42 \times 10^{-3}$
witness-aleatório & witness-reverso	$2.82 \times 10^{-2}$
witness-aleatório & aleatório	$1.73 \times 10^{-2}$
aleatório & witness-reverso	$8.97 \times 10^{-1}$

Tabela 6 – Resultados do teste Exato de Fisher entre cada um dos pares de currículos para a taxa de desistência.

Par de Currículos	$p$	$d$
witness-exato & witness-aleatório	$8.85 \times 10^{-8}$	1.210
witness-exato & witness-reverso	$2.79 \times 10^{-18}$	1.788
witness-exato & aleatório	$2.96 \times 10^{-14}$	1.618
witness-exato & equidistante	$1.77 \times 10^{-2}$	0.481
equidistante & witness-aleatório	$8.97 \times 10^{-5}$	0.795
equidistante & witness-reverso	$3.24 \times 10^{-17}$	1.659
equidistante & aleatório	$5.92 \times 10^{-12}$	1.351
witness-aleatório & witness-reverso	$5.05 \times 10^{-7}$	0.717
witness-aleatório & aleatório	$3.14 \times 10^{-3}$	0.456
aleatório & witness-reverso	$1.17 \times 10^{-2}$	0.282

Tabela 7 – Resultados dos testes de Mann-Whitney entre os pares de currículos para o número de puzzles resolvidos antes de desistir do experimento.

Par de Currículos	$p$	$d$
witness-exato & witness-aleatório	$1.11 \times 10^{-1}$	0.262
witness-exato & witness-reverso	$7.22 \times 10^{-4}$	0.594
witness-aleatório & witness-reverso	$4.95 \times 10^{-2}$	0.358

Tabela 8 – Resultados dos testes de Mann-Whitney entre os pares de currículos para o número de puzzles que os participantes não-desistentes resolveram.

currículo.

O Kruskal-Wallis para o número de backtracks nas instâncias do currículo resultou em um valor de 20.56,  $p$ -valor de  $3.86 \times 10^{-4}$ , e effect size ( $d$ ) de 0.50. A Tabela 12 apresenta os  $p$ -valores das comparações pareadas para o número de backtracks nas instâncias do currículo. O Kruskal-Wallis para o número de backtracks nas instâncias de teste resultou em uma estatística de 1.82,  $p$ -valor de  $7.68 \times 10^{-1}$ , e effect size ( $d$ ) de 0.17. O Kruskal-Wallis para o número de backtracks na primeira instância de teste resultou em um valor de 1.28,  $p$ -valor de  $8.63 \times 10^{-1}$ , e effect size ( $d$ ) de 0.19. Já que os testes entre todos os currículos

Currículo	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12
witness-exato	0.99	0.97	0.97	0.91	0.88	0.93	0.81	0.81	0.76	0.35	0.91	0.80
witness-aleatório	0.79	0.84	0.80	0.80	0.80	0.84	0.89	0.89	0.89	0.32	0.88	0.68
witness-reverso	0.48	0.42	0.50	0.76	0.76	0.92	0.96	0.96	1.00	0.24	0.78	0.72
aleatório	0.75	0.80	0.70	0.86	0.84	0.86	0.84	0.82	0.86	0.45	0.86	0.82
equidistante	1.00	0.98	0.86	0.79	0.86	0.88	0.89	0.82	0.21	0.39	0.86	0.75

Tabela 9 – Fração dos participantes não-desistentes que resolveu cada puzzle por currículo.

Par de Currículos	$p$	$d$
witness-exato & witness-aleatório	$6.31 \times 10^{-1}$	0.084
witness-exato & witness-reverso	$9.15 \times 10^{-1}$	0.019
witness-exato & aleatório	$3.73 \times 10^{-1}$	0.164
witness-exato & equidistante	$3.73 \times 10^{-3}$	0.494
equidistante & witness-aleatório	$3.27 \times 10^{-2}$	0.409
equidistante & witness-reverso	$8.07 \times 10^{-3}$	0.529
equidistante & aleatório	$1.28 \times 10^{-1}$	0.305
witness-aleatório & witness-reverso	$5.70 \times 10^{-1}$	0.111
witness-aleatório & aleatório	$7.22 \times 10^{-1}$	0.072
aleatório & witness-reverso	$2.81 \times 10^{-1}$	0.224

Tabela 10 – Resultados dos testes de Mann-Whitney entre os pares de currículos para o número de tentativas nos puzzles de teste que os participantes não-desistentes performaram.

Currículo	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12
witness-exato	0.64 (1.52)	1.79 (1.77)	0.79 (2.50)	3.24 (6.37)	5.23 (11.56)	2.75 (4.49)	4.39 (6.27)	4.23 (6.27)	5.17 (7.61)	7.96 (7.79)	1.48 (3.48)	3.21 (6.50)
witness-aleatório	8.61 (11.64)	3.70 (5.49)	3.89 (5.74)	4.18 (6.67)	3.96 (11.03)	2.59 (4.62)	2.73 (5.96)	1.55 (2.49)	2.11 (4.30)	6.73 (6.28)	1.89 (4.43)	2.93 (3.53)
witness-reverso	12.12 (10.14)	6.18 (5.62)	4.10 (5.46)	3.18 (4.84)	2.16 (6.24)	1.62 (5.47)	0.20 (0.99)	0.18 (0.68)	0.56 (1.13)	7.96 (6.18)	2.98 (6.04)	2.00 (3.47)
aleatório	8.64 (6.86)	4.57 (5.39)	5.43 (0.09)	3.86 (8.77)	2.93 (4.54)	2.30 (4.93)	2.45 (4.83)	1.95 (4.33)	0.84 (2.77)	7.52 (7.14)	0.70 (1.92)	2.11 (3.31)
equidistante	0.70 (1.56)	4.40 (3.94)	6.75 (6.71)	5.28 (5.83)	3.30 (6.02)	3.86 (6.96)	1.42 (2.84)	1.82 (3.47)	13.37 (12.91)	5.23 (5.56)	0.53 (1.90)	1.58 (2.69)

Tabela 11 – Número de tentativas apresentado como M (DP) por puzzles e currículo.

para as instâncias de teste e a primeira instância de teste não foram estatisticamente significantes, nós não fizemos comparações pareadas para estes casos. A Tabela 13 mostra a média do número de backtracks para cada puzzle e currículo.

Nenhum dos resultados de tempo foram estatisticamente significantes. O Kruskal-Wallis para a média de tempo que os participantes gastaram para resolver as instâncias do currículo retornou um valor de 8.18,  $p$ -valor de 0.08, e effect size ( $d$ ) de 0.24. O Kruskal-Wallis para a média de tempo que os participantes gastaram para resolver as instâncias de teste retornou um valor de 7.69,  $p$ -valor de 0.10, e effect size ( $d$ ) de 0.23. O Kruskal-Wallis para a média de tempo que os participantes gastaram para resolver a primeira instância de teste retornou um valor de 4.65,  $p$ -valor de 0.32, e effect size ( $d$ ) de 0.09.

Par de Currículos	$p$	$d$
witness-exato & witness-aleatório	$5.38 \times 10^{-2}$	0.341
witness-exato & witness-reverso	$2.51 \times 10^{-1}$	0.206
witness-exato & aleatório	$1.84 \times 10^{-1}$	0.245
witness-exato & equidistante	$2.54 \times 10^{-5}$	0.787
equidistante & witness-aleatório	$1.38 \times 10^{-2}$	0.476
equidistante & witness-reverso	$1.20 \times 10^{-3}$	0.659
equidistante & aleatório	$1.41 \times 10^{-2}$	0.504
witness-aleatório & witness-reverso	$2.70 \times 10^{-1}$	0.216
witness-aleatório & aleatório	$7.38 \times 10^{-1}$	0.067
aleatório & witness-reverso	$5.97 \times 10^{-1}$	0.110

Tabela 12 – Resultados dos testes de Mann-Whitney entre os pares de currículos para o número de backtracks nos puzzles dos currículos que os participantes não-desistentes performaram.

Currículo	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12
witness-exato	0.08 (0.31)	0.19 (0.50)	0.07 (0.57)	0.28 (1.11)	0.41 (1.13)	0.28 (1.20)	1.39 (2.46)	1.95 (4.28)	2.96 (4.72)	10.55 (15.35)	1.03 (2.16)	3.85 (7.79)
witness-aleatório	2.46 (5.41)	1.20 (3.48)	0.96 (3.21)	1.00 (1.80)	1.07 (2.63)	0.89 (2.31)	0.84 (2.24)	1.16 (2.43)	1.29 (4.89)	10.57 (13.77)	0.50 (1.28)	3.89 (6.14)
witness-reverso	2.40 (2.66)	3.70 (6.62)	0.94 (2.14)	0.22 (0.57)	0.78 (1.99)	0.14 (0.49)	0.00 (0.00)	0.04 (0.27)	0.06 (0.23)	8.64 (11.40)	0.88 (1.55)	1.86 (3.15)
aleatório	1.36 (2.41)	1.11 (2.33)	2.27 (9.68)	1.25 (2.74)	1.30 (2.97)	1.02 (2.27)	1.05 (2.76)	2.52 (10.25)	0.77 (2.33)	11.20 (13.35)	0.66 (1.64)	3.27 (5.52)
equidistante	0.04 (0.18)	0.44 (1.21)	0.95 (2.47)	0.51 (0.95)	0.40 (0.91)	1.30 (2.55)	1.18 (2.39)	2.12 (3.10)	13.75 (17.65)	7.67 (9.80)	0.63 (1.11)	3.04 (5.86)

Tabela 13 – Número de backtracks apresentados como M (DP) por puzzle e currículo.

Currículo	US	CA	GB	BR	AU	DE	IN	HK	FR	Outros	Não-especificado
witness-exato	28	6	0	5	4	2	1	3	2	16	56
witness-aleatório	28	16	8	2	3	2	1	2	0	17	32
witness-reverso	36	8	7	3	3	1	5	1	2	19	53
aleatório	20	11	7	4	3	4	3	0	1	22	51
equidistante	24	6	9	4	2	2	1	0	0	13	48
Total	136	47	31	18	15	11	11	6	5	87	240

Tabela 14 – Número de participantes por país e currículo antes de contar os desistentes.

## A.4 Dados Demográficos

As Tabelas 14 e 15 mostram o país de origem dos participantes de cada um dos currículos antes e depois de remover os desistentes, respectivamente. Mostramos apenas os 9 países com o maior número de participantes; os restantes são representados na coluna ‘Outros’. A Tabela 16 e 17 apresenta a distribuição de gênero dos participantes por currículo antes e depois de remover os desistentes, respectivamente. A Tabela 18 e 19 apresenta a distribuição de idade dos participantes por currículo antes e depois de remover os desistentes, respectivamente. A Tabela 20 e 21 apresenta a distribuição de nível de escolaridade dos participantes por currículo antes e depois de remover os desistentes, respectivamente.

As Tabelas 22, 23, 24 e 25 apresentam os resultados para a frequência em que os participantes jogam jogos e resolvem puzzles, antes e depois de remover os desistentes.

Currículo	US	CA	GB	BR	AU	DE	IN	HK	FR	Outros	Não-especificado
witness-exato	19	2	0	4	2	2	0	2	1	9	34
witness-aleatório	15	9	4	1	1	1	1	0	0	12	13
witness-reverso	13	5	3	1	1	0	4	0	1	6	16
aleatório	6	3	1	2	3	1	0	0	0	10	18
equidistante	12	4	8	2	0	0	0	0	0	8	22
Total	65	23	16	10	7	4	5	2	2	45	103

Tabela 15 – Número de participantes por país e currículo depois de contar os desistentes.

Currículo	Masculino	Feminino	Outro	Não-especificado
witness-exato	40	46	0	37
witness-aleatório	45	42	1	23
witness-reverso	45	54	4	35
aleatório	39	55	2	30
equidistante	36	41	1	31
Total	205	238	8	156

Tabela 16 – Número de participantes por gênero e currículo antes de contar os desistentes.

Currículo	Masculino	Feminino	Outro	Não-especificado
witness-exato	25	27	0	23
witness-aleatório	26	20	1	9
witness-reverso	16	21	2	11
aleatório	15	22	0	7
equidistante	23	22	1	11
Total	105	112	4	61

Tabela 17 – Número de participantes por gênero e currículo depois de contar os desistentes.

## A.5 Interface Gráfica do Usuário

As Figuras 6-12 mostram capturas de tela da interface gráfica dos usuários no nosso experimento.

Currículo	18–35	36–55	56+	Não-especificado
witness-exato	68	17	1	37
witness-aleatório	63	17	5	26
witness-reverso	79	21	2	36
aleatório	75	11	3	37
equidistante	54	18	1	32
Total	339	84	12	172

Tabela 18 – Número de participantes por idade e currículo antes de contar os desistentes.

Currículo	18–35	36–55	56+	Não-especificado
witness-exato	37	14	1	23
witness-aleatório	34	9	3	10
witness-reverso	30	9	0	11
aleatório	28	6	0	10
equidistante	33	8	0	16
Total	162	46	4	70

Tabela 19 – Número de participantes por idade e currículo depois de contar os desistentes.

Currículo	Pre High School	High-School	College	Graduate School	Professional School	PhD	Postdoctoral Training
witness-exato	3	11	64	20	5	16	4
witness-aleatório	4	19	55	18	4	11	0
witness-reverso	5	18	72	23	6	8	6
aleatório	4	24	62	17	7	11	1
equidistante	3	17	49	21	5	12	2
Total	19	89	302	99	27	58	13

Tabela 20 – Número de participantes por nível de escolaridade e currículo antes de contar os desistentes.

Currículo	Pre High School	High-School	College	Graduate School	Professional School	PhD	Postdoctoral Training
witness-exato	2	5	34	14	3	14	3
witness-aleatório	0	9	24	12	3	8	0
witness-reverso	2	7	26	1	1	5	1
aleatório	0	7	25	5	5	2	0
equidistante	2	5	24	16	2	6	2
Total	6	33	133	55	14	35	6

Tabela 21 – Número de participantes por nível de escolaridade e currículo depois de contar os desistentes.

Currículo	Nunca	Muito Raramente	Raramente	Ocasionalmente	Frequentemente	Muito Frequentemente	Sempre	Não-especificado
witness-exato	2	7	5	25	19	14	8	41
witness-aleatório	1	8	6	20	20	15	10	31
witness-reverso	3	8	10	27	24	21	5	40
aleatório	1	6	5	22	25	15	9	43
equidistante	0	5	4	25	16	14	6	39
Total	7	34	30	119	104	79	38	196

Tabela 22 – Número de participantes por frequência de jogar jogos e currículo antes de contar os desistentes.

Currículo	Nunca	Muito Raramente	Raramente	Ocasionalmente	Frequentemente	Muito Frequentemente	Sempre	Não-especificado
witness-exato	0	6	5	14	14	7	3	26
witness-aleatório	0	2	2	10	12	10	8	12
witness-reverso	1	3	4	13	7	7	2	13
aleatório	0	2	3	10	9	5	2	13
equidistante	0	4	3	16	10	5	3	16
Total	1	17	17	63	52	34	18	80

Tabela 23 – Número de participantes por frequência de jogar jogos e currículo depois de contar os desistentes.

Currículo	Nunca	Muito Raramente	Raramente	Ocasionalmente	Frequentemente	Muito Frequentemente	Sempre	Não-especificado
witness-exato	2	8	14	29	20	6	1	43
witness-aleatório	0	9	11	36	17	7	0	31
witness-reverso	3	13	13	32	25	11	2	39
aleatório	1	12	11	28	18	11	2	43
equidistante	0	8	16	23	14	9	0	39
Total	6	50	65	148	94	44	5	195

Tabela 24 – Número de participantes por frequência de resolver puzzles e currículo antes de contar os desistentes.

Currículo	Nunca	Muito Raramente	Raramente	Ocasionalmente	Frequentemente	Muito Frequentemente	Sempre	Não-especificado
witness-exato	1	6	10	17	12	2	1	26
witness-aleatório	0	3	8	16	12	5	0	12
witness-reverso	1	4	5	18	5	4	0	13
aleatório	0	6	5	8	7	4	1	13
equidistante	0	7	7	14	7	6	0	16
Total	2	26	35	73	43	21	2	80

Tabela 25 – Número de participantes por frequência de resolver puzzles e currículo depois de contar os desistentes.

## Do you solve puzzles like a robot?

You're about to take a test on LabintheWild. Your participation allows us to learn more about how people around the world solve puzzles. The results from your test will also tell you something about yourself!

*Please read the following information carefully before proceeding.*

**Why we are doing this research?** We are trying to understand how solving puzzles in different orders affects a person's ability to solve new puzzles.

**What you will have to do?** We will ask you to solve puzzles. More instructions about this task will follow.

**What you will get out of it?** We will give you feedback on how your results compare to those of other participants and of a computer. The final results from this experiment will be posted on our [webpage](#) and presented at conferences. Please email [\[redacted\]](#) if you would like us to send you a copy of the results when they are ready.

The experiment is not designed to benefit you and has no known risks, but you may enjoy it and enjoy comparing your results with those of other participants.

**Privacy and Data Collection?** We will not ask you for your name. Any data that we collect will be securely stored on our servers. We will delete your data if you do not complete the study or you tell us not to use it, but we cannot remove your data once you have received your results.

Everyone is welcome to try to solve our puzzles, but we will only use the data of those who are at least 18 years old in our research.

**Duration?** You will solve puzzles for about 15 minutes.

The plan for this study has been reviewed by a Research Ethics Board at [\[redacted\]](#). If you have questions about your rights or how research should be conducted, you can call [\[redacted\]](#). For international calls include the [\[redacted\]](#).



By ticking this box, you are agreeing to be in the study. Be sure that questions you have about the study have been answered and that you understand what you are being asked to do. You may contact the researcher if you think of a question later. You are free to leave the experiment at any time, and refusing to be in the experiment or stopping participation will involve no penalty or loss of benefits to which you are otherwise entitled.



Figura 6 – Formulário de consentimento.

Please tell us a bit about yourself.  
 Fields marked with a \* are required

Have you taken this test before? \*

What is your gender?

How old are you?

What is your native language? \*

Have you lived in more than one country?

Please add countries in order from the country you were born in to the most recent one you are living in

Country

What is the highest level of education that you have received or are pursuing? \*

How many years of education have you completed, starting from primary school?

How often do you play games?

How often do you solve puzzles?



Figura 7 – Questionário demográfico.

Goal: You want to connect the black circle to the pulsing circle.

Use a keyboard and hit the RIGHT button, or if your keyboard doesn't have it, use the "d" button.

You can use a mouse or your finger to tap on the virtual keys below the puzzle as well.

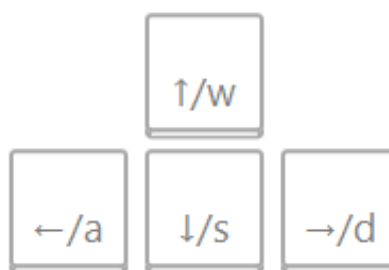
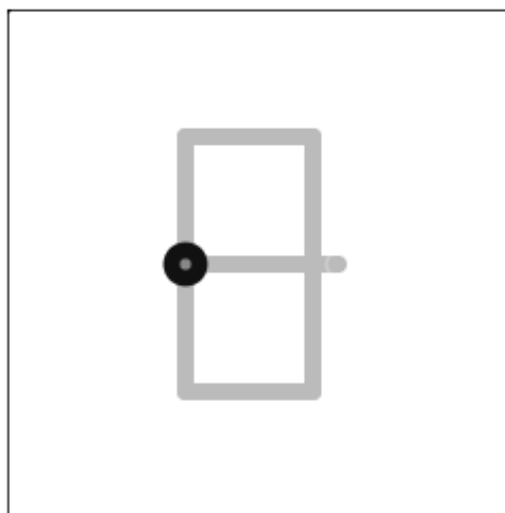


Figura 8 – Instruções em como usar os controles do teclado.

## Nice job!

Let's get started!

Try to solve each puzzle as quickly as you can.

We aren't going to tell you how to solve them because we want to see how quickly you can figure this out. If you get stuck, you can skip a puzzle.

We'll give you another chance to try each puzzle at the end.

Remember

- It's okay to take breaks between puzzles.
- It's okay to skip puzzles. We will give you another chance to do them after we've given you your results.



Figura 9 – Instruções finais antes de começar a resolver as instâncias do currículo.

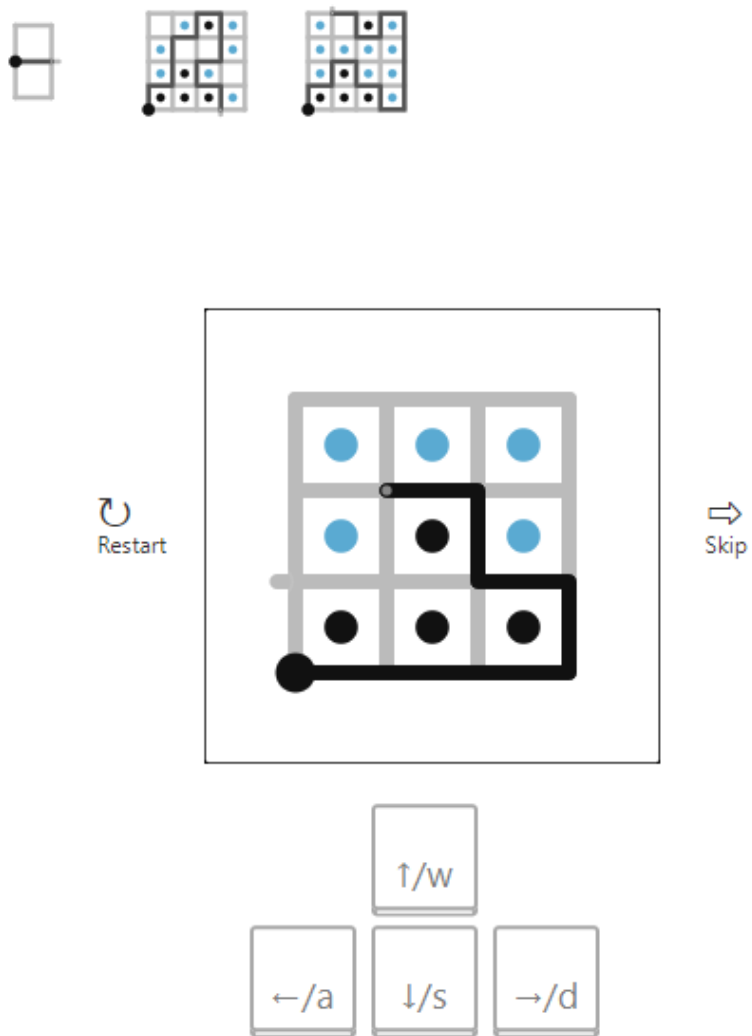


Figura 10 – Interface de um dos puzzles, com alguns puzzles já resolvidos aparecendo no topo.

## Thank you for your participation!

**Before you continue to your results, please let us know a little bit more about yourself.**

Have you played The Witness (the puzzles came from this game) before? \*

- No  
 Yes

**We'd also love to hear what you thought of the test!**

Do you have any comments for the researcher? Questions, Suggestions, or Concerns?

Is there any reason we should exclude your results from our study?

- No  
 Yes

If so, why? (optional) For example: during the test, you changed your monitor settings, received help from another person or source...etc

Did you encounter any technical difficulties or interruptions during this study? If so, what?

Did you cheat or in any way provide false information? If yes, how?



Figura 11 – Interface da página de feedback geral para nosso experimento.

## Let's see how you did!



### You're Almost Like the Robot!

With a score of 88.1, you solved almost as many puzzles as a robot would. You might just be a highly logical thinker.

### How did you do?

You solved 12 puzzles and hit restart on puzzles 5 times.

### Why are we doing this?

We don't know the best puzzle order to give to people to help them learn how to solve puzzles and other types of problems. Figuring this out will help us to know how to support people's learning so that learning materials can be better designed.

### If you enjoyed this study, there's more!

1. Visit and follow on Facebook to learn about new studies: [facebook.com/LabintheWild](https://facebook.com/LabintheWild)
2. Go to the [Lab in the Wild site](#) for more studies.
3. Share this study with your friends!



### Retry previous puzzles

If you'd like to play any puzzles you've skipped or just to retry any of them, click on the arrow button on the bottom right.



Figura 12 – Página onde damos Feedback ao participante.