

## Seleção de Caminho de Aprendizagem para Grupo de Usuários: uma Abordagem baseada em Perfil

Mayara S. O. Castro<sup>1</sup>, Lizandra L. S. B. Silva<sup>1</sup>, Samuel B. J. Souza<sup>2</sup>,  
Giuseppe F. Neto<sup>1</sup>, Lucas A. Santos<sup>1</sup>, Rafael F. L. Mello<sup>1</sup>, Péricles B. C. Miranda<sup>1</sup>

<sup>1</sup>Departamento de Estatística e Informática  
Universidade Federal Rural de Pernambuco (UFRPE)

<sup>2</sup>Centro de Informática  
Universidade Federal de Pernambuco (UFPE)

**Abstract.** *Identifying learning paths is a key issue for effective learning. This article proposes an intelligent method to optimize the process of selection of learning paths for a group of individuals taking into account multiple criteria: satisfaction of the team members and time spent in the accomplishment of the activity. The method was evaluated regarding the performance, being compared to the exhaustive and random approaches; And regarding the pedagogical aspect, being compared with random and self-selected methods. The results showed the potential of the proposed method from the computational point of view as well as the pedagogical point of view.*

**Resumo.** *A identificação de caminhos de aprendizagem é uma questão chave para a aprendizagem eficiente. Este artigo propõe um método inteligente para otimizar o processo de seleção de caminhos de aprendizagem para um grupo de indivíduos, levando em conta múltiplos critérios: satisfação dos membros da equipe e tempo gastos na realização da atividade. O método foi avaliado quanto ao desempenho, sendo comparado às abordagens exaustiva e aleatória; e quanto ao aspecto pedagógico, sendo comparado com os métodos aleatório e auto-selecionado. Os resultados obtidos mostraram o potencial do método proposto tanto do ponto de vista computacional quanto do pedagógico.*

### 1. Introdução

Os recentes avanços no campo da *e-learning* possibilitaram a utilização de ambientes virtuais para alcançar estudantes sem limites espaciais e temporais [Voss et al. 2013]. No entanto, ainda existem dificuldades enfrentadas nesses ambientes como: falta de motivação dos alunos, acompanhamento individual do professor para a grande quantidade de estudantes e utilização de todos os recursos disponíveis. Um dos motivos para esses problemas ocorrerem é que a aprendizagem não se dá apenas através da disponibilização de recursos de aprendizagem, mas também da identificação de caminhos de aprendizagem interessantes para cada estudante [dos Santos et al. 2015].

Um Caminho de aprendizagem é um conjunto de unidades de aprendizagem (por exemplo, materiais de estudo) que determina o caminho pelo qual o aluno deve seguir, em termos de conhecimento a ser adquirido ou aperfeiçoado, para que o mesmo realize tarefas relacionadas a essas competências [Li et al. 2009]. Em muitas situações, alunos

precisam participar de atividades em grupo, como desenvolver um site ou escrever um artigo. Assim, para concluir a atividade, cada membro do grupo precisa desenvolver habilidades e aprender novos conteúdos para realizar suas tarefas. Portanto, é fundamental conhecer o perfil de cada membro do grupo e identificar um caminho de aprendizagem eficiente e efetivo para cada um, de modo que ajude a equipe terminar a atividade [Xie et al. 2017].

Diante deste cenário, esse trabalho propõe um método automático para a sugestão de caminhos de aprendizagem para membros de um grupo, levando em conta os seus perfis. Para isso, o problema em questão é tratado como sendo uma tarefa de otimização com múltiplos critérios em um contexto combinatório. Neste artigo dois objetivos são considerados: maximização da satisfação dos membros do grupo, fazendo com que cada membro alcance seus resultados de aprendizagem almejados; e minimizar o tempo necessário para a conclusão da atividade em equipe. Foram realizadas avaliações quantitativa e pedagógica no método proposto. Para a realização da análise quantitativa, comparamos o método com as abordagens exaustiva e aleatória, usando um conjunto de dados de valores simulados. Para a avaliação pedagógica, foi realizado um experimento real com estudantes de um *Massive Open Online Course (MOOC): Desenvolvendo um e-Commerce com Python 3 e Django*. Comparamos o método com duas abordagens tradicionais para definição de caminhos de aprendizagem: aleatória e auto-organizada. Os resultados constataram que, tanto do ponto de vista de desempenho quanto do pedagógico, o método proposto é eficiente e capaz de formar caminhos de aprendizagem adequados para os grupos, considerando os critérios de satisfação e tempo.

Este artigo está organizado da seguinte forma: a Seção 2 introduz o contexto do problema de definição de caminhos de aprendizagem e os principais trabalhos relacionados. A Seção 3 descreve o método desenvolvido. A Seção 4 apresenta a metodologia experimental utilizada para avaliar o método proposto. A Seção 5 apresenta os resultados alcançados. Finalmente, a Seção 6 destaca as conclusões e trabalhos futuros.

## 2. Contexto do Problema

A tarefa de definir caminhos de aprendizagem torna-se difícil, devido à complexidade na qual as características e condições iniciais do grupo interagem entre si [Xie et al. 2017]. Um caminho de aprendizagem inadequado, quando atribuído à um indivíduo, pode levar a consequências negativas [Xie et al. 2017], tais como: falta de interesse, atraso na entrega da atividade e má formação de competências [Cardoso et al. 2016, Belacel et al. 2014]. Embora seja um problema da área de educação, abordagens computacionais têm sido desenvolvidas e aplicadas neste contexto.

O trabalho de [Steiner and Albert 2007] propõe, a partir de redes conceituais dinâmicas, um mecanismo responsável por entregar planos de estudo personalizados e individuais. [Feng et al. 2010] definiram métodos para encontrar os caminhos de aprendizagem coletivos, partindo da análise de cada indivíduo do grupo. Diferentemente dos trabalhos anteriores, [Hwang et al. 2010] considerara a descoberta de caminhos de aprendizagem como sendo um problema de otimização, na qual a solução ideal seria a maximização da aprendizagem dos alunos. Já [Madhour and Wentland Forte 2008] utilizaram um algoritmo inspirado em colônia de formigas (ACO) baseado no modelo do usuário e na experiência de outros alunos, proporcionando caminhos de aprendizagem personaliza-

dos. Também vale mencionar o trabalho de [Belacel et al. 2014], que aplicou um modelo de programação inteira binária para a otimização de caminhos de aprendizagem. Por fim, destaca-se o trabalho desenvolvido por [Xie et al. 2017] que utilizou um método exaustivo para encontrar caminhos ótimos para grupos de indivíduos levando em conta o perfil de cada membro. Este último trabalho foi o primeiro a definir caminhos de aprendizagem em um cenário coletivo.

Diferentemente dos estudos apresentados, propomos um método evolucionário para a seleção automática de caminhos de aprendizagem para um grupo, considerando múltiplos critérios.

### 3. Formulação do Método

O problema de definição de caminhos de aprendizagem para um grupo alunos demanda um espaço de soluções de dimensão elevada, com múltiplos critérios a serem considerados. Diante disto, decidimos utilizar um algoritmo genético multi-objetivo (MOGA).

#### 3.1. Otimização Multi-Objetivo

A otimização multi-objetivo (MO) visa encontrar soluções que satisfaçam mais de um critério ao mesmo tempo. Considere um vetor  $\vec{C} = (C_1, C_2, \dots, C_n)$  de objetivos, onde  $n$  é o número de objetivos a serem considerados na otimização. Deste modo, cada solução possui um vetor  $\vec{f} = (f_1, f_2, \dots, f_n)$ , onde cada valor (*fitness*) de  $\vec{f}$  representa a avaliação da solução em cada critério pertencente a  $\vec{C}$ . Como neste trabalho são adotados dois critérios na otimização  $\vec{C} = (C_1, C_2)$ , cada solução possui um vetor  $\vec{f} = (f_1, f_2)$ . Como cada solução possui um vetor de *fitness*, a comparação entre diferentes soluções se dá da seguinte forma: dadas duas soluções distintas  $S_1$  e  $S_2$ , onde  $f_{(S_1)}$  representa o vetor de *fitness* de  $S_1$  e  $f_{(S_2)}$  representa o vetor de *fitness* de  $S_2$ ;  $S_1$  domina  $S_2$  se  $f_{(S_1)}$  é melhor que  $f_{(S_2)}$  em ao menos um *fitness*, e não é pior em nenhum outro. Deste modo, existem três casos possíveis: 1)  $S_1$  domina  $S_2$ , 2)  $S_1$  é dominada por  $S_2$ , ou 3)  $S_1$  é incomparável em relação à  $S_2$ . Este último caso ocorre quando  $S_1$  e  $S_2$  vencem em pelo menos um critério. O conjunto de soluções incomparáveis (ou não-dominadas) é conhecido como *Pareto Front*. Deste modo, algoritmos de otimização multi-objetivo tentam encontrar o *Pareto Front* ótimo, ou seja, aquele conjunto de soluções não-dominadas que melhor satisfazem os critérios considerados. No MOGA, uma população inicial de indivíduos (soluções) é definida, e esta passa, iterativamente, por processos evolucionários, tais como avaliação, seleção, cruzamento e mutação de indivíduos. Após um determinado número de evoluções, o *Pareto front* resultante, que reflete as melhores soluções, é gerado como saída do algoritmo.

#### 3.2. Representação do Indivíduo

Um indivíduo (cromossomo) em um GA é tomado como uma solução para o problema em que se quer resolver. No problema em questão, objetiva-se alocar um conjunto de unidades de aprendizagem  $\vec{U} = (U_1, U_2, \dots, U_t)$ , onde  $t$  é o número de unidades, entre os membros do grupo. Neste caso, cada cromossomo representa uma possível alocação de unidades de aprendizagem para um grupo de membros, onde cada membro do grupo é encarregado de suas unidades. Definindo-se formalmente, um cromossomo seria um vetor de membros do grupo  $\vec{I} = (M_1 : \vec{u}_1, M_2 : \vec{u}_2, \dots, M_l : \vec{u}_l)$ , onde  $l$  é o número de membros do grupo; e cada membro  $M_i$  possui um vetor de unidades de aprendizagem

alocadas a ele  $\vec{v}_i$ . É importante salientar que diferentes membros não podem possuir a mesma unidade de aprendizagem, e que não há membros sem unidades de aprendizagem alocadas.

Como neste trabalho o processo de otimização do caminho de aprendizagem para um grupo é baseado no perfil de cada membro, é importante definir quais características são relevantes para a descrição dos membros e das unidades. Vamos considerar que existam  $k$  características importantes para a realização das unidades contidas em  $\vec{U}$ . Deste modo, cada membro do grupo  $M_i, i = 1..l$ , possui um vetor de preferências  $\vec{P}$  e um vetor de habilidades  $\vec{H}$ , onde cada valor desses vetores estão relacionados a cada uma das  $k$  características.  $\vec{P}$  armazena valores reais no intervalo  $[0,1]$  equivalente ao interesse do membro em relação a certos temas; E  $\vec{H}$  armazena valores reais no intervalo  $[0,1]$  equivalente às habilidades do membro em relação a certos temas. Esses dois vetores são usados para definir o perfil de cada membro do grupo em relação às unidades de aprendizagem consideradas.

Assim como cada membro do grupo precisou ser descrito através das  $k$  características, cada unidade de aprendizagem contida em  $\vec{U}$  também o foi. Cada unidade possui: 1) a duração de tempo ( $T$ ) em que um aluno com valor 0 de habilidade naquela unidade em questão gastará para realizá-la; e 2) vetor de habilidades requeridas  $\vec{HR}$ , atribuindo-se a cada habilidade um valor no intervalo  $[0,1]$ , equivalente ao quanto a atividade a necessita para ser concluída.

### 3.3. Avaliação do Indivíduo

Uma das funções mais importantes do MOGA é a avaliação das soluções (cromossomo) da população. É através dessa avaliação que será determinado o quão boa é a solução, e se será selecionada para as próximas etapas na evolução do algoritmo. Neste trabalho, dois objetivos são considerados na avaliação das soluções, sendo: o tempo, que deve ser minimizado, visto que as atividades devem ser concluídas de maneira a reduzir o tempo total gasto; E a preferência ou satisfação, que deve ser maximizada, garantindo que cada estudante fique responsável por uma atividade que tenha maior afinidade. Os cálculos necessários para a avaliação das soluções serão apresentados a seguir.

**Cálculo da Satisfação ( $ST$ ):** Com relação à avaliação da satisfação de cada membro do grupo, foi criada uma matriz  $MA_{membros,unidades}$  que armazena o somatório do módulo da diferença, entre os valores de mesmo critério, de  $\vec{P}$  (vetor de preferências do membro) e  $\vec{HR}$  (vetor de habilidades requeridas pela unidade de aprendizagem). Cada valor de  $MA$  é calculado da seguinte forma:

$$MA_{a,b} = \sum_{k=1}^{n^{\circ}\_habilidades} |M_a.P_k - U_b.HR_k|, \quad (1)$$

onde  $M_a.P_k$  é o  $k$ -ésimo valor do vetor de preferências  $\vec{P}$  do membro  $M_a$ , e  $U_b.HR_k$  é o  $k$ -ésimo valor do vetor de habilidades requeridas  $\vec{HR}$  da unidade de aprendizagem  $U_b$ . Dessa forma, um valor de preferência de cada estudante em relação à cada unidade de aprendizagem é gerado. Quanto mais próximo de zero o valor obtido for, maior será o grau de satisfação em trabalhar com aquela unidade de aprendizagem. Uma vez que  $MA$  está preenchida, esta é utilizada para calcular o valor (*fitness*) de satisfação total do grupo

(*ST*). *ST* nada mais é que a soma da satisfação individual (*Si*) de cada componente em relação às unidades de aprendizagem alocadas a cada um deles. Deste modo, o valor de *ST* é obtido através da equação 2. Diante disto, quanto menor o valor de *ST*, maior a satisfação total do grupo. Vale salientar que o valor de *ST* é normalizado entre [0,1].

$$ST = \sum_{j=1}^{n^{\circ}\text{ _membros}} Si_j. \quad (2)$$

**Cálculo do Tempo (*TT*):** O método proposto otimiza as soluções em relação ao tempo considerando duas situações: minimização do tempo geral de finalização das unidades e aproximando a duração total gasta por cada membro nas unidades alocadas a cada um. O objetivo de levar em conta estas duas situações, é fazer com que haja o estudo das unidades de aprendizagem em menor tempo, garantindo uma menor probabilidade de ocorrência de sobrecargas entre os membros do grupo.

Na avaliação do tempo, as habilidades dos membros foram consideradas. Como já dito anteriormente, cada valor em  $\vec{H}$  (vetor de habilidades de cada membro) é definido entre o intervalo [0,1]. Deste modo, quanto maior for o valor de uma habilidade, as unidades de aprendizagem que requererem aquele tipo habilidade, tendem a serem finalizadas em menor tempo. Assim, sendo possível a geração de uma matriz,  $MT_{membros,unidades}$ , encarregada de definir a previsão do tempo restante em que cada membro levaria para finalizar cada unidade. O cálculo de cada valor de *MT* é realizado da seguinte forma:

$$MT_{a,b} = (1 - M_a.H_b \times U_b.T), \quad (3)$$

onde  $M_a.H_b$  é o *b*-ésimo valor do vetor de habilidades  $\vec{H}$  do membro  $M_a$ , e  $U_b.T$  é o tempo de um aluno com valor 0 de habilidade na unidade de aprendizagem  $U_b$ . Uma vez preenchida, *MT* é utilizada para calcular o valor (*fitness*) de tempo total do grupo (*TT*). *TT* nada mais é do que a soma do tempo gasto de cada membro (*Ti*) na finalização de suas unidades de aprendizagem. Deste modo, o valor de *TT* é obtido assim:

$$TT = \sum_{j=1}^{n^{\circ}\text{ _membros}} Ti_j. \quad (4)$$

Como havíamos dito previamente, todos os membros do grupo precisam trabalhar de forma igualitária para evitar a sobrecarga dos demais alunos. Como não há garantia de que um valor de *TT* baixo signifique que todos os membros do grupo trabalharam de forma igualitária, decidimos criar uma penalidade. Para isto, foi realizado um somatório das diferenças entre o tempo individual de cada membro do grupo:

$$Penalidade = \sum_{a=1}^{n^{\circ}\text{ _membros}} \sum_{b=a}^{n^{\circ}\text{ _membros}} |Ti_a - Ti_b|. \quad (5)$$

Deste modo, quanto menor o valor da *Penalidade*, maior é a aproximação do tempo gasto, entre os estudantes, para a finalização de suas unidades. Diante da importância desta penalidade, incorporamos ao cálculo de *TT*, o valor da *Penalidade*<sup>2</sup>,

resultando em  $TT = TT + Penalidade^2$ . Em situações em que o valor de  $TT$  é baixo e há sobrecarga de trabalho, a *Penalidade* aumenta o valor do *fitness* fazendo com que a solução tenha menos chances de ser escolhida para a próxima geração. Vale salientar que o valor de  $TT$  é normalizado entre  $[0,1]$ .

#### 4. Metodologia Experimental

Neste trabalho, duas análises foram realizadas: uma que avalia o *desempenho* e outra que avalia o ponto de vista *pedagógico*. Na primeira análise, o desempenho do método proposto foi comparado com um método exaustivo e um aleatório (ambos em suas versões multi-objetivo, levando em consideração a satisfação e o tempo). Todos os métodos usaram um conjunto de dados (características e unidades de aprendizagem) de valores simulados (uniformemente distribuídos). Vale salientar que as características consideradas são as mesmas descritas na seção 4.2. Por questões práticas de comparação dos resultados, decidimos, ao invés de comparar os *Paretos* obtidos por cada método, selecionar apenas uma solução de cada *Pareto Front*. Esta prática é muito comum em análises de métodos multi-objetivo, e geralmente utiliza-se o método de *Borda Count* [Orouskhani et al. 2017]. Este método utiliza uma estratégia de ranqueamento por objetivo, de modo que a solução do *Pareto Front* que estiver melhor ranqueada é selecionada. O objetivo da análise de desempenho é verificar se o método proposto é capaz de encontrar um bom caminho de aprendizagem em tempo hábil.

Para a realização da análise pedagógica, foi realizado um experimento com estudantes do MOOC: *Desenvolvendo um e-Commerce com Python 3 e Django*, em fase de conclusão do curso e elaboração do projeto final. A turma considerada possui 120 alunos: graduandos, profissionais e pesquisadores com experiência e conhecimentos diversos, fatores importantes nesta pesquisa. Os 120 alunos foram divididos, pelos professores, em 24 grupos de 5 membros. A cada grupo foi atribuída a tarefa de desenvolver um *e-commerce em Python 3 e Django*, com o auxílio de 25 unidades de aprendizagem (5 unidades por aluno), sendo cada uma um material e estudo relativo a um determinado assunto. Além disso, embora os projetos de cada grupo apresentem o mesmo nível de complexidade e requisitos similares, cada equipe é responsável por um projeto diferente para evitar cópia.

Uma vez apresentado o estudo de caso, o método proposto é comparado com duas abordagens tradicionais para definição de caminhos de aprendizagem: aleatória e auto-selecionada. Como existem ao todo 24 grupos de 5 alunos, cada método foi aplicado sobre 8 grupos diferentes. O método aleatório atribui, a cada aluno, unidades de aprendizagem aleatórias. Já no método auto-selecionado, os alunos escolhem as suas próprias unidades. Os grupos atrelados a cada abordagem serão avaliados pedagogicamente, ou seja, em relação aos seus desempenhos (notas) obtidos em seus projetos. O objetivo é avaliar se, na prática, o método proposto é capaz de melhorar o desempenho dos seus grupos, quando comparados aos das demais abordagens. O cálculo da nota final do grupo é a média entre a nota do projeto, e a média das notas de cada aluno do grupo. Deste modo, projetos que não foram entregues por completo e/ou alunos que não tenham contribuído com a realização do projeto, sejam motivo para penalização da nota do grupo.

##### 4.1. Unidades de Aprendizagem

Como pode-se ver na Tabela 1, existem 25 unidades que abrangem diversos tipos de conhecimento: *design, configuração, codificação, modelagem, teste e deploy*. Cada unidade

está associada a um conjunto de requisitos. Deste modo, os alunos devem resolver os requisitos do sistema que estiverem associados às suas 5 unidades.

**Tabela 1. Unidades de Aprendizagem relativas ao projeto de um e-commerce**

Unidades de Aprendizagem		
Orientação a Objetos em Python	django.contrib.messages	Teste de Performance
Configuração do Django	Custom Template Tags	Teste de Views
Configuração do Banco de Dados	Custom User e Custom User Model	Teste de Forms e E-mail
Desenvolvimento de Templates	Class-based views para o Fórum	Teste de Models com model-mommy
App Courses	Signal para envio de emails	Deploy no Heroku
Model Objects	Ajustando Design do Dashboard	Prototipação do design da aplicação
URL's	Modelagem do Banco de Dados	Html e CSS
Views: Página, Listagem e Login	Mapeamento dos Dados	
Formulários	Testes Unitários	

## 4.2. Características Consideradas

Neste trabalho foram adotados dois aspectos para definir o perfil dos alunos e o perfil das unidades de aprendizagem: preferência e habilidades em certos assuntos. Estes aspectos foram selecionados com base em uma revisão da literatura, nas características particulares da população estudada e nas unidades de aprendizagem consideradas (ver Tabela 1). Diante disto, 7 características foram selecionadas ( $k = 7$ ) relativas à: 1) *design*, 2) configuração, 3) desenvolvimento de programas, 4) modelagem, 5) teste de software, 6) *deploy*, e 7) liderança. A escolha destas características é baseada no fato de que é interessante que cada grupo tenha membros responsáveis por atividades que sabem e gostam de fazer.

A extração de características dos alunos envolvidos neste experimento se deu através da aplicação de um questionário, e da análise do desempenho de cada aluno no curso e em outros cursos da plataforma online. O questionário consistiu em um conjunto de perguntas de resposta única e de múltipla escolha sobre tópicos relacionados às características.

## 4.3. Configurações

O método proposto adotou o MOGA chamado *Non-Dominated Sorting Genetic Algorithm* (NSGA II) adaptado para o problema em questão. O método utilizou a seleção baseada em *roleta*, onde 40% dos indivíduos da população são clonados e mantidos na geração seguinte. Os outros 60% dos indivíduos são utilizados pelo operador de cruzamento. O operador de mutação foi aplicado na população em cada geração com uma probabilidade de mutação para cada indivíduo  $\theta = 0.2$  e uma probabilidade de mutação para cada cromossomo  $\theta = 0.15$ . O tamanho populacional,  $TAM = 50$ , bem como o número máximo de gerações,  $n\_gen = 500$ , foram definidos empiricamente. O método aleatório adotou o mesmo número de soluções candidatas que as utilizadas para o método proposto. Os métodos aleatório e proposto foram executados 30 vezes para gerar um desempenho médio. Todos os métodos foram implementados em Python usando o ambiente Microsoft Visual Studio 2010 e todas as execuções foram realizadas em um laptop com Intel Core i7-5600U com 4M Cache, 8GB RAM e 3.20 GHz.

## 5. Resultados

Esta seção apresenta resultados provenientes de duas análises diferentes. A primeira leva em conta o aspecto desempenho (quantitativo), enquanto a segunda, o aspecto pedagógico.

### 5.1. Análise de Desempenho

Nesta análise, o desempenho do método proposto foi comparado com o da abordagem exaustiva e aleatória. A Tabela 2 mostra o tempo médio de execução de cada método (medido em segundos) e o valor de *fitness* médio alcançado em cada objetivo (satisfação e tempo de realização da atividade). Por uma questão de simplicidade, o número de membros por grupo foi de 5 em todos os casos. Além disso, o desempenho de cada método foi avaliado variando-se o número de unidades de aprendizagem.

**Tabela 2. Análise comparativa do desempenho entre as abordagens exaustiva, aleatória e proposta.**

Unidades	Exaustivo			Aleatório			Proposta		
	Tempo	ST	TT	Tempo	ST	TT	Tempo	ST	TT
10	0.49	0.21	0.23	0.59	<b>0.25</b>	<b>0.23</b>	0.62	<b>0.21</b>	<b>0.23</b>
15	90.45	0.16	0.19	7.89	0.28 (↓)	0.27 (↓)	9.87	<b>0.19</b>	<b>0.22</b>
20	878.43	0.13	0.17	12.34	0.35 (↓)	0.34 (↓)	15.76	<b>0.16</b>	<b>0.21</b>
25	7349.2	0.09	0.13	32.12	0.37 (↓)	0.37 (↓)	40.98	<b>0.14</b>	<b>0.18</b>
30	18216.1	0.03	0.06	59.76	0.41 (↓)	0.42 (↓)	72.34	0.11 (↓)	0.16 (↓)
50	N/A	N/A	N/A	187.36	0.52	0.47	210.39	0.07	0.13
100	N/A	N/A	N/A	420.29	0.63	0.51	480.74	0.07	0.10

Como pode-se ver, o tempo de execução necessário para o método exaustivo avaliar todas as soluções possíveis aumenta drasticamente a medida que o número de unidades de aprendizagem também aumenta. Por outro lado, o tempo de execução do método proposto e aleatório aumenta quase que linearmente. É importante mencionar que o método exaustivo alcança melhores valores de *fitness* à medida que aumenta a quantidade de unidades de aprendizagem. Isso se deve ao fato de que mais unidades de aprendizagem, mais combinações de caminhos podem ser formados. O mesmo não acontece com o método aleatório pois, quanto maior o número de soluções possíveis, menores são as chances de encontrar boas combinações de caminhos por sorte.

Quanto ao desempenho do método proposto, vale destacar que mesmo não alcançando valores ótimos (iguais aos do exaustivo), o método foi capaz de encontrar boas soluções em tempo de execução menor, até quando a quantidade de unidades de aprendizagem aumenta. Para realizar uma comparação justa entre os métodos, aplicamos o teste de hipótese *t-student*, considerando 95% de significância. Aqueles valores de *fitness*, apresentados na Tabela 2, que estiverem em negrito significam que são estatisticamente iguais aos do exaustivo. Em contrapartida, os valores com o símbolo ↓ são estatisticamente inferiores aos do exaustivo. A comparação estatística realizada não levou em consideração os valores para 50 e 100 alunos, uma vez que os resultados do método exaustivo não foram obtidos por questões de custo computacional.

Como pode-se ver, a maioria dos resultados alcançados pelo método aleatório, para cada objetivo, foram superados estatisticamente pelos resultados da abordagem exaustiva. Apenas no cenário mais simples que a abordagem aleatória teve bons resultados. Por outro lado, os valores de *fitness* obtidos pelo método proposto foram estatisticamente iguais aos do método exaustivo nos cenários de 10, 15, 20 e 25 alunos. Embora os valores para 30 unidades tenham sido estatisticamente piores que o do exaustivo, os resultados foram bons e alcançados em bem menos tempo. Talvez, com mais gerações, melhores resultados seriam alcançados. Com relação aos resultados para 50 e 100 alunos, o método proposto foi capaz de melhorar ainda mais os valores de *fitness* dos dois



objetivos. Quando comparado à abordagem aleatória, o método proposto a superou estatisticamente em todos os objetivos, na maioria dos cenários.

## 5.2. Análise Pedagógica

Os grupos, atrelados a cada método, tiveram seu desempenho pedagógico (nota do grupo) avaliado. A nota do grupo é a composição da nota do projeto e da nota de cada membro da equipe (ver seção 4). A Tabela 3 apresenta os resultados pedagógicos colhidos através da aplicação do método proposto, aleatório e auto-selecionado, em 24 turmas de 5 estudantes (8 turmas por método). Estão explicitados na Tabela 3, por método, a média e desvio padrão das notas dos projetos dos grupos; a média e o desvio padrão das notas dos indivíduos dos grupos; e a média e o desvio padrão das notas dos grupos. De acordo com os professores, os projetos feitos por equipes com caminhos de aprendizagem definidos a partir do método entregaram o projeto com todos os requisitos. A média das notas dos projetos entregues é 8.8 com um desvio padrão de 1.1, ou seja, o projeto com menor nota ainda assim atingiu nota acima de 7.0 (nota necessária para aprovação). Quanto às notas individuais, nota-se que a média das notas dos membros também é alta. Isso comprova a efetividade da contribuição de cada membro das equipes em seus projetos, e a entrega das funcionalidades requisitadas. Como consequência, a média final dos grupos atendidos pelo método proposto foi bastante satisfatória.

**Tabela 3. Média das notas dos grupos e individuais de cada abordagem.**

	N. Alunos	Notas dos Projetos	Notas Individuais	Notas dos Grupos
Proposta	40	8.8 ( $\pm 1.1$ )	8.4 ( $\pm 0.6$ )	8.6 ( $\pm 0.85$ )
Aleatória	40	5.1 ( $\pm 0.98$ )	6.4 ( $\pm 0.47$ )	5.75 ( $\pm 0,72$ )
Auto-selecionada	40	7.5 ( $\pm 3.5$ )	6.2 ( $\pm 3.1$ )	6.85 ( $\pm 3.3$ )

Os grupos que adotaram a abordagem auto-selecionada obtiveram 7.5 como média das notas nos projetos, mas com 3.5 de desvio padrão, ou seja, as notas dos projetos das equipes tiveram grande variação. O mesmo aconteceu com as notas individuais, onde a média foi de 6.2 e o desvio padrão de 3.1. Este cenário, em que as notas apresentam grande variação, mostra o desbalanceamento da contribuição dos alunos. De acordo com os professores, alguns grupos, que adotaram a abordagem auto-selecionada, não entregaram o projeto com todos os requisitos. Como consequência, a média do grupo foi penalizada, resultando em 6.85. Por fim, os resultados mostram que a abordagem aleatória não foi capaz de atingir resultados competitivos frente aos métodos proposto e auto-selecionado. A média das notas dos projetos daqueles grupos que adotaram a abordagem aleatória é 5.1 e a média das notas individuais é 6.4, ambas com baixo desvio padrão. De acordo com o relato dos professores, algumas equipes não chegaram a entregar todos os requisitos do projeto. Diante de tudo isto, a nota média dos grupos, que adotaram esta abordagem, é 5.75.

## 6. Conclusão e Trabalhos Futuros

Este trabalho apresenta um método automático para a definição de caminhos de aprendizagem para membros de um grupo, considerando critérios como a satisfação dos membros da equipe e o tempo gasto pelo grupo na realização de uma atividade. Os resultados mostraram que o método proposto foi capaz de atingir desempenho estatisticamente similar ao do método exaustivo na maioria dos cenários, mas, com custo de tempo quase linear.

Além disso, o método superou a abordagem aleatória na maioria dos critérios. Em relação à análise pedagógica, foi constatado que grupos que utilizaram a recomendação de caminhos de aprendizagem sugeridos pelo método proposto obtiveram melhores notas na atividade dada quando comparados com os demais grupos. Futuramente, pretendemos considerar outras características para ajudar na melhor definição do perfil do aluno. Além disso, objetivamos investigar e incluir novos critérios no processo de otimização, além de satisfação e tempo de entrega da atividade (considerados neste trabalho).

## Referências

- Belacel, N., Durand, G., and Laplante, F. (2014). A binary integer programming model for global optimization of learning path discovery. In *EDM (Workshops)*.
- Cardoso, I. C., Rissoli, V., Moreira, T., and Borges, V. (2016). Construção de ambientes interativos de apoio à reflexão docente aplicados à teoria da aprendizagem significativa. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, volume 5, page 637.
- dos Santos, H., Cechinel, C., Araújo, R., and Brauner, D. (2015). Recomendação de objetos de aprendizagem utilizando filtragem colaborativa: Uma comparação entre abordagens de pré-processamento por meio de clusterização. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 26, page 1127.
- Feng, X., Xie, H., Peng, Y., Chen, W., and Sun, H. (2010). Groupized learning path discovery based on member profile. In *International Conference on Web-Based Learning*, pages 301–310. Springer.
- Hwang, G.-J., Kuo, F.-R., Yin, P.-Y., and Chuang, K.-H. (2010). A heuristic algorithm for planning personalized learning paths for context-aware ubiquitous learning. *Computers & Education*, 54(2):404–415.
- Li, Q., Lau, R. W., Wah, B. W., Ashman, H., Leung, E. W., Li, F., and Lee, V. (2009). Guest editors' introduction: Emerging internet technologies for e-learning. *IEEE Internet Computing*, 13(4):11–17.
- Madhour, H. and Wentland Forte, M. (2008). Personalized learning path delivery: Models and example of application. In *Intelligent Tutoring Systems*, pages 725–727. Springer.
- Orouskhani, M., Teshnehlab, M., and Nekoui, M. A. (2017). Evolutionary dynamic multi-objective optimization algorithm based on borda count method. *International Journal of Machine Learning and Cybernetics*, pages 1–29.
- Steiner, C. M. and Albert, D. (2007). Personalising learning through prerequisite structures derived from concept maps. In *International Conference on Web-Based Learning*, pages 43–54. Springer.
- Voss, G. B., Nunes, F. B., Herpich, F., and Medina, R. D. (2013). Ambientes virtuais de aprendizagem e ambientes imersivos: um estudo de caso utilizando tecnologias de computação móvel. In *Anais do Simpósio Brasileiro de Informática na Educação*, volume 24, page 12.
- Xie, H., Zou, D., Wang, F. L., Wong, T.-L., Rao, Y., and Wang, S. H. (2017). Discover learning path for group users: A profile-based approach. *Neurocomputing*.